UNIVERSITY OF ALBERTA

STABILITY ENHANCEMENT OF MULTI–MACHINE POWER SYSTEMS BY HYBRID NEURAL FUZZY–LOGIC CONTROL

By

Stéphan Zefeng Ao

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Department of Electrical Engineering

Edmonton, Alberta

Fall 1996

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file *Votre référence*

Our file *Notre référence*

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-612-18011-5

Canada

UNIVERSITY OF ALBERTA

Library Release Form

Name of Author: Stéphan Zefeng Ao

Title of Thesis: **STABILITY ENHANCEMENT OF MULTI–MACHINE POWER SYSTEMS BY HYBRID NEURAL FUZZY–LOGIC CONTROL**

Degree: Doctor of Philosophy

Year This Degree Granted: 1996

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Signed: Stephan

Address: Box #308
Herschel, Saskatchewan,
S0L 1L0 CANADA

Date: October 4, 1996

UNIVERSITY OF ALBERTA

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **STABILITY ENHANCEMENT OF MULTI–MACHINE POWER SYSTEMS BY HYBRID NEURAL FUZZY–LOGIC CONTROL** submitted by Stéphan Zefeng Ao in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

K. E. Bollinger (Supervisor)

J. C. Salmon

R. E. Rink

D. G. Fisher

O. P. Malik (External Examiner)

Date: 27/09/96

To my parents,

brothers and sisters

for your love and pride

ACKNOWLEDGMENT

I am deeply indebted to my supervisor, Professor K. E. Bollinger, for his constant encouragement, guidance, and support given throughout the research reported in this thesis.

I would like to thank Emeritus Professor R. J. Fleming, Emeritus Professor Maria Green for their encouragement and support in the past few years.

I am grateful to the Wiebe family, without their love, trust, and support, I would not be able to finish this work.

Last, but by no means least, I am truly indebted to the people I have met and the people I have heard of but not met yet, for your life–time inspiration, **"The Best You Can Be."**

I would like to express my deep appreciation to NSERC and the University of Alberta for providing financial assistance throughout this research.

ABSTRACT

This thesis presents a hybrid modeling technique for designing intelligent controllers for multimachine power systems using artificial neural networks(ANNs) and fuzzy logic(FL). ANNs are unstructured numerical estimators that can learn, generalize and adapt. FL is a structured representation of knowledge. An ANN–FL controller is designed in Chapter 5 for speed–tracking control of a DC motor. Simulation studies have shown that the newly–designed ANN–FL controller outperforms fixed parameter PID control when the DC motor experiences a load or a reference speed change.

This technique is then utilized to design supplementary excitation control for a two–machine infinite–bus power system. Two ANN–FL power system stabilizers (PSSs), one using speed deviation as input and the other using accelerating power, are designed. As ANN–FL PSSs have no dynamics themselves, and require only simple algorithms, they respond instantly to changes in the state variables and are robust to changes in the operating conditions. This research has led to an innovative design technique for supplementary excitation control of synchronous generators. An ANN–FL PSS synthesized using this technique can be employed on different generators without customized design.

Finally, an innovation to programming transient simulation has been achieved in this research. Difficulties in handling generator saliencies have been overcome in designing a simulation algorithm implemented by using the Transient Stability Simulation Package(TSSP) under SIMULINK.

TABLE OF CONTENTS

LIBRARY RELEASE FORM
ACKNOWLEDGMENT
ABSTRACT
TABLE OF CONTENTS
LIST OF TABLES
LIST OF FIGURES
LIST OF PRINCIPAL SYMBOLS

1. **Introduction** 1
    1.1 Power System Stability 1
    1.2 Stability of Synchronous Generators 5
        1.2.1 General Concepts 5
        1.2.2 Conventional PSS control 6
        1.2.3 Adaptive PSS Control 9
        1.2.4 Intelligent Control Systems 11
    1.3 Statement of Objectives 21
    1.4 Outline of the Thesis 23

2. **Artificial Neural Networks** 26
    2.1 Introduction 26
    2.2 A Biological Neuron 28
    2.3 Mathematical Model of A Neuron 39
    2.4 Actuation Functions 34
    2.5 Neural Network Architectures 37
    2.6 Learning Rules 38
    2.7 Multilayer Perceptrons 39

3. **Fuzzy Sets and Fuzzy Reasoning** 43
3.1 Introduction 43
3.2 Fuzzy Sets 44
3.3 Fuzzy Set Operations 45
3.4 Linguistic Variables 49
3.5 Fuzzy Logic and Approximate Reasoning 50
3.6 Inference Methods 52
3.7 Defuzzification 58

4. **Hybrid ANN-Fuzzy Logic Modeling** 60
4.1 Introduction 60
4.2 Fuzzy Logic Control System 61
4.3 Architecture of ANN-Fuzzy Logic Modeling 64
4.4 Learning Rules 70

5. **ANN–FLC Application: Speed–Tracking Control of A DC Motor** 76
5.1 Introduction 76
5.2 Problem Formulation 77
5.3 Design of An ANN–FL Controller 82
5.3.1 Introduction 82
5.3.2 Structure of the ANN–FL Controller 83
5.4 Performance Evaluation 94
5.4.1 Introduction 94
5.4.2 Step Response with Different Reference Speeds 95
5.4.3 Different Disturbances 97
5.4.4 Trajectory Control 101

6. **Modeling of Composite Power Systems** 103
6.1 Introduction 103
6.2 Modeling of Power Components 104

6.2.1 Synchronous Machines 104
6.2.2 Excitation Systems 112
6.2.3 Prime Movers 116
6.2.4 Conventional PSS 119
6.2.5 Power System Loads 121
6.2.6 Induction Motors 123
6.2.7 DC Transmissions 129
6.3 System Disturbances 133

7. **A Transient Stability Simulation Package (TSSP) under Simulink** 135
7.1 TSSP under DOS 135
7.2 Necessity of Improvement 139
7.3 TSSP under Simulink 141
7.3.1 Introduction 141
7.3.2 Basics of SIMULINK 142
7.3.3 Building Blocks 143
7.3.4 Model Construction 156
7.4 TSSP's Potential with Simulink 157

8. **Stability Enhancement of A Multimachine Power System Using An ANN-FL PSS** 160
8.1 Introduction 160
8.2 Stability Analysis 164
8.2.1 A Multimachine Power System 164
8.2.2 Stability Studies 165
8.3 Design of Two ANN-FL PSSs 171
8.3.1 General Comments 171
8.3.2 ANN–FL PSS Using Speed Deviation 172

8.3.3 ANN–FL PSS Using Accelerating Power 176
8.4 Performance Evaluation of the ANN-FL PSS 180
8.4.1 General Comments 180
8.4.2 Step Reference Voltage Change 180
8.4.3 Step Load Change 184
8.4.4 Fault Conditions 185
8.4.5 ANN–FL PSS Using Accelerating Power 193
8.5 Summary 194

9. **Conclusions** 195
9.1 The Initiative 195
9.2 Realization 197
9.3 Significance of the Work 198
9.4 Future Work 200

References 201
Appendix A Saturated Reactance and Time Constants 212
Appendix B M–File find_vif.m 216
Appendix C Specifications of the Two–Machine Infinite–Bus System 220
Appendix D TSSP Block Library 222

LIST OF TABLES

Table 5.1 Rule Base of the ANN–FL Controller....88
Table 5.2 Training Data as Obtained from Simulation Studies....93
Table 8.1 Load Flow Results: Node Power....168
Table 8.2 Load Flow Results: Branch Power....168
Table 8.3 Sources of Data Collection....177

LIST OF FIGURES

2.1 A Biological Neuron 30
2.2 Response to A Stimulus 30
2.3 Mathematical Model of the Neuron 31
2.4 Nonlinear Model of A Neuron 33
2.5 Final Model of the Neuron 33
2.6 Symmetric Hard Limit Activation Function 35
2.7 Tanh Sigmoid Activation Function 36
2.8 Log Sigmoid Activation Function 36
2.9 A Feedforward Multilayer Neural Network 38
2.10 Output Neuron j 42

3.1 Fuzzy Set Operations 46
3.2 Support, Core and Boundary 49
3.3 Membership Functions of Temperature 50
3.4 Type 1 Fuzzy Reasoning 56
3.5 Type 2 Fuzzy Reasoning 56

4.1 Fuzzy Logic Control System 63
4.2 ANN-FL Modeling, Types 1 & 2 Reasoning 66
4.3 ANN-FL Modeling, Types 3 & 4 Reasoning. For Type 4 reasoning, The Input Vector x Has to Be Fed to the Function Vector g 67

5.1 Schematic Representation of A DC Motor 80
5.2 Block Diagram of A DC Motor 80
5.3 Simulation Model of A DC Motor 81
5.4 Speed Response of the DC Motor with No Feedback Control 83
5.5 Membership Functions for State Variable $\Delta\omega / \Delta t$ 87
5.6 Membership Functions for State Variable $\Delta\omega$ 87
5.7 Membership Functions for Output Variable ΔV 88

5.8 An ANN-FL Controller for DC Motor Control 91
5.9 Membership Functions for State Variable $\Delta\omega / \Delta t$ 91
5.10 Membership Functions for State Variable $\Delta\omega$ 92
5.11 Membership Functions for Output Variable ΔV 92
5.12 RMSE Curve vs. Epochs 92
5.13 Step Sizes vs. Epochs 93
5.14(a) Response of PID and ANN–FL Control to A Step Input of 90 rad/s....96
5.14(b) Response of PID and ANN–FL Control to A Step Input of 130 rad/s....96
5.15 Performance Comparison of ANN–FL vs. PID Control, with A Periodical Load Disturbance Applied....98
5.16 Window Graphic Showing Transient Details in Figs. 5.15(b) & (c)....99
5.17(a) A Periodical Field Voltage Disturbance 99
5.17(b) Performance of PID Control Responding to the Disturbance of Fig. 5.17(a)....100
5.17(c) Performance of ANN–FL Control Responding to the Disturbance of Fig. 5.17(a)....100
5.18 Window Graphic Showing Transient Details in Figs. 5.17(b) & (c)....100
5.19 Trajectory Tracking Control of the DC Motor by (a) PID or (b) ANN–FL Controller....102

6.1 Pictorial Representation of A Synchronous Machine 106
6.2 Block Diagram of the Synchronous Generator Model 109
6.3 Block Diagram of DC Type 1 Excitation System 114
6.4 Block Diagram of A Digital Excitation System 114
6.5 Alternator–Rectifier Excitation System....115
6.6 Detailed Representation of Hydrogovernor Turbine System 117
6.7 Equivalent Representation of Hydrogovernor Turbine System....117
6.8 Speed Governing System for Steam Turbines 118

6.9 Block Diagram of A Steam Turbine 118
6.10 IEEE Standard PSS Model 120
6.11 IEEEST PSS Using Net Accelerating Power As Input 120
6.12 IEEE PSS Using Generated Electric Power As Input 120
6.13 Induction Motor 125
6.14 Block Diagram of An Induction Motor 129
6.15 Schematic Diagram of A DC Line 131
6.16 Equivalent Circuit of A HVDC Link 131
6.17 Converter Representation 132

7.1 Overall Structure of the TSSP 140
7.2 Simulation Model of A Synchronous Generator 145
7.3 Simulation Model of the DC Type 1 Excitation System 146
7.4 Simulation Model of the Hydrogovernor Turbine System 149
7.5 Speed Governing System for Steam Turbines 150
7.6 Tandem–Compound Double–Reheat Steam Turbine 151
7.7 Simulation Model of IEEE Standard PSS 152
7.8 Simulation Model of IEEEST PSS 153
7.9 Simulation Model of IEEESN PSS 153
7.10 Simulation Model of An Induction Motor Model 154
7.11 Single–Line Diagram of the One–Machine System 158
7.12 Simulation Model of the One–Machine System 159

8.1 Single–Line Diagram of the Two Machine Infinite Bus System 165
8.2 Frequency Response Showing the Two Oscillatory Modes 169
8.3 Simulation Model of the Study System 169
8.4(a) Field Voltage Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1 170
8.4(b) Field Voltage Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1 170

8.5 Generator Angle Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1 171
8.6 Triangular Membership Function 177
8.7 Trapezoidal Membership Function 177
8.8 Membership Functions for State Variable $\Delta\omega / \Delta t$ 178
8.9 Membership Functions for State Variable $\Delta\omega$ 178
8.10 Membership Functions for Output Variable ΔV 178

8.11(a) Machine Angle Response to A 4% Step Voltage Change 182
8.11(b) Reactive Power Response to A 4% Step Voltage Change 182
8.11(c) Terminal Voltage Response to A 4% Step Voltage Change 183
8.11(d) Field Voltage Response to A 4% Step Voltage Change 183
8.12(a) Machine Angle Response to A 0.25pu Step Load Change 186
8.12(b) Electric Power Response to A 0.25pu Step Load Change 186
8.12(c) Terminal Voltage Response to A 0.25pu Step Load Change 187
8.12(d) Controller Response to A 0.25pu Step Load Change 187
8.13(a) Rotor Angle Response to A Three–Phase to Ground Fault at Bus G#1 for 6–Cycles, with One Controller 189
8.13(b) Electric Power Response to A Three–Phase to Ground Fault at Bus G#1 for 6–Cycles, with One Controller 189
8.14(a) Rotor Angle Response to A Three–Phase to Ground Fault at Bus G#1 for 6–Cycles, with Two Controllers 190
8.14(b) Terminal Voltage Response to A Three–Phase to Ground Fault at Bus G#1 for 6–Cycles, with Two Controllers 190
8.15 Rotor Angle Response to A Three–Phase to Ground Fault on the Line 1–4 for 6–Cycles 192
8.16 Rotor Angle Response to A Three–Phase to Ground Fault on the Line 2–3 for 6–Cycles 192
8.17 Angle Response to the 3–Phase to Ground Fault on Line 1–2, with the ANN–FL PSS Using Accelerating Power As Input Signal 193

LIST OF PRINCIPAL SYMBOLS

Rotor angles of synchronous generators are with reference to the infinite–bus synchronous frame. Parameters of transmission lines are in per unit (pu) based on the system base 100 MVA. Machine impedance is in per unit based on its own rated capacity. All time constants are in seconds. Machine inertial H is also in seconds based on the machine capacity.

Abbreviations:

deg = degree

rad = radian

s = second

ANN = Artificial Neural Network

AI = Artificial Intelligence

FL = Fuzzy Logic

PR = Probabilistic Reasoning

AVR = Automatic Voltage Regulator and Exciter

AGC = Automatic Generation Control

CPSS = Conventional Power System Stabilizer

STC = Self–Tuning Control

MRAC = Model Reference Adaptive Control

RLS = Recursive Least Squares

CMAC = Cerebeelar Model Arithmetic Computer

ATO = Automatic Train Operation at Sandai, Japan

FFT = Fast Fourier Transform

G = Generator

DC = Direct Current

AC = Alternating Current

PID = Proportional, Integral, and Derivative

PE = Processor/Processing Element

HVDC = High Voltage Direct Current

A/D = Analog/Digital

D/A = Digital/Analog

RAM = Random Access Memory

DSP = Digital Signal Processing

TGR = Transient Gain Reduction

TSSP = Transient Stability Simulation Package

FDLF = Fast Decoupled Load Flow

STCC = Short Circuit Calculation

WPLOT = Plotting Program

Symbols:

$\rightarrow$ = mapping

$\int$ = fuzzy union(continuous), integral

$\sum$ = fuzzy union(discrete)

$\cup, \cap, -$ = union, intersection, complement, respectively

$\in$ = belong to

δ = machine rotor angle (radian or degree), propagation error

ω = machine angular speed

ω_0 = synchronous speed (rad/s)

η = learning rate

φ = activation function

θ = threshold, mechanical angle

E = voltage, error function

e = error function

E' = voltage behind transient reactance

E'' = subtransient voltage

f = frequency

i, I = current

P, Q = active and reactive power

v, V = voltage

H = machine inertial constant (s)

k = saturation factor

K = gain of transfer function

D = machine damping coefficient

τ, T = time constant

t = target output

ϕ = flux

x = reactance, input to neural network

R = resistance

T = torque, term set

$\underline{X}$ = fuzzy set

U = universe of discourse

μ_x = membership function

α = firing strength

u = control variable

v = net internal activation

L = Laplace transform

A, B = label of membership function

Superscripts:

$'$ = transient

$''$ = subtransient

(0) = unsaturated value

l = layer

Subscripts:

a = armature

e = electrical

m = mechanical

$D, Q,$ or

d, q = direct and quadrature axes

l = leakage, power losses

t, T = terminal

s = supplementary signal

f, fd = field

i, j = index

ref = reference

err = error

min = minimum

max = maximum

o = open circuit, initial value

p = pattern, product

ad, aq = mutual

x, y = x and y coordinates

w = washout, weight

0 = initial value

Operators:

$\wedge$ = intersection

$\oplus$ = bounded product

$\cap$ = drastic product

$x \cdot y$ = algebraic product

$\bullet$ = fuzzy composition

Δ = small change

$\sum$ = summation

$\times$ = multiplication

$\|$ = parallel

s = Laplace operator

CHAPTER 1

INTRODUCTION

This chapter presents an introduction to the problem of power system stability and its control. Basic concepts, classification and definition of related terms used in power system stability analysis are described. A review of three categories of control strategies for synchronous generator control is conducted. They include conventional power system stabilizer(PSS) control, adaptive PSS control and intelligent control systems. The objectives of the thesis are then summarized. Finally an outline of the thesis is presented.

1.1 POWER SYSTEM STABILITY

An electric power system consists of a large number of generators, transmission lines, and various loads. It is characterized by dynamic phenomena of which stability and control are dominant. To maintain a successful operation of a power system, there are three functional requirements that must be satisfied simultaneously, namely, service continuity, voltage and frequency stability, and economic efficiency. Of these three requirements, system stability and control is of primary importance.

Synchronous generators connected through a transmission network must run in parallel, that is, at the same instantaneous speed. Automatic

control systems are installed to keep the rotating speed difference among the generators within narrow bounds. If one generator momentarily runs faster than another, the angular position of its rotor relative to that of the slower generator will continue to advance as long as the speed difference exists, and its generated voltage will likewise advance in phase relative to the voltage of the slower machine. The resultant phase difference shifts a load from the slower machine to the faster machine, tending to reduce the speed difference.

This shifting process is a nonlinear function of the difference in rotor angles. When this difference in rotor angles exceeds a upper limit, load shifting reverses and speed difference increases instead. This is referred as the loss–of–synchronism phenomena. Loss of synchronism can occur between one machine and the rest of the system or between groups of machines. Power system stability may take different forms, depending on the state of the system. During fault conditions, the major concern is whether the system can stay interconnected. The loss of interconnection will only increase instability. On the other hand, during normal operation, the task is to keep voltage and frequency fluctuation within an acceptable error.

Power system stability can be classified into short–term and long–term stability. The former can be further categorized by steady–state stability and transient stability. Steady–state stability is the stability following a small disturbance for which the equations that describe the dynamics of the system can be linearized for the purpose of analysis. A system is said to be steady–

state stable for a particular operating condition, following a small disturbance, if the system can reach another steady–state condition that may or may not be identical or close to the pre–disturbance operating condition of the power system. A system is said to be transient stable for a particular operating condition and for a specified large disturbance if, following that disturbance, it can reach an acceptable steady state operating condition[1–1].

Load plays an important role in system stability. Lightly loaded power transmission lines force leading power factors on generators, making them more unstable. Steam and diesel plants can only reduce their loads to about 50% full load; below that, they begin to shut down. Daily start up and shutdown cycles create tremendous thermal stress and reduce the useful life, especially of steam plants[1–1]. On the other hand, heavy loads require huge amounts of spinning reserve to meet possible plant outages. For this reason, interconnected tie lines should never trip on overload. They may stay connected to their thermal limits.

In a given system, load or demand is the major independent variable that initiates changes. Everything else in the system is designed to meet the demand with reliability, quality and economy. When demand changes, corresponding changes in frequency and voltage occur. As the equipment in a system is designed to operate at a certain voltage and frequency, these two quantities must be controlled when the above said changes occur. This is why there are always controllers even on old power equipment.

As our society's consumption of electrical energy has increased steadily, power systems over large geographic areas have been interconnected to meet this demand and to provide a reliable electrical power service. At the same time, sophisticated control equipment and protection schemes have been added to power systems to enhance its stability. As a result the analysis of power system stability has become more difficult. Fortunately, advanced mathematical modeling and better computation facilities have made it possible to solve the complicated problem of stability investigations. In this thesis, the focus is placed on short-term stability studies, using artificial neural networks(ANNs) and fuzzy logic(FL) theory.

Synchronous generators are to supply sufficient electric energy to meet the demand in the system. The objective of generator control is to maintain the terminal voltage and frequency at some predetermined values. This is accomplished by two principal control loops, one for frequency control and the other for voltage control. In long-term stability, frequency control is accomplished by a speed-governor through a turbine that drives the generator; voltage control is accomplished by a voltage regulator through an excitation system that provides the bridge for converting mechanical power into electrical power in the generator.

In short-term stability, i.e., transient stability, the working mechanism of the control is different. As transient stability accounts for only several seconds following the disturbance, the speed-governing and turbine system is

not capable of following the changes the system is experiencing. For practical reasons it is detrimental to the equipment to open and close the steam valve so often and so fast when the transient is oscillatory. In the voltage control loop, as fast–acting excitation systems are used with high gain, when there is a contingency, such as a three–phase to ground fault, the voltage dips. The excitation system will raise the field voltage to the upper ceiling and boost the reactive power of the generator so that the voltage level can be brought back to normal.

At the same time, this fast–acting excitation brings in another problem. For situations with heavy load or weak transmission system, the high gain AVR introduces negative damping. This means that after the first swing in the transient, sustained oscillations can occur. This is harmful because the power transfer over transmission lines has to decrease and the stability integrity of the system is in danger. This was why in the 60's power systems engineers started installing additional controllers called power system stabilizer(PSS) to suppress sustained low frequency oscillations.

1.2 Stability of Synchronous Generators

1.2.1 General Concepts

Power system stability is principally concerned with the stability of synchronous generators and tie–line stability in a system. As the load increases, more generators are continuously added to the system. To achieve

economic benefits, generators of larger sizes and high reactance have been designed. As a result, more sophisticated control designs are required to provide compensating effects offsetting the reductions in stability margins inherent from such designs. Research on this subject has received great attention since the 70's[1–2]. Over the past three decades, continuing attention has been focused on the effects of excitation control on the damping of oscillations. These supplementary excitation controllers, i.e., power system stabilizers(PSSs), have been added to counteract the effects of high gains of automatic voltage regulators(AVRs). These AVRs usually have high gains and are fast. This way transient system stability can enhanced. An adverse effect of the fast–acting AVRs is a reduction in the damping.

As the size of synchronous generators increases, more spinning reserve is required in the system. To achieve economical operation, local power systems are connected together through tie–lines to reduce the requirement of high spinning reserves. This practice has given rise to another problem that is apart from local oscillations. It is possible that inter–area oscillations can occur following certain disturbances in certain geographical areas.

1.2.2 Conventional PSS control

Excitation control can increase the stability limits of synchronous generators. Exciters need to have a high gain to perform this function. To achieve a high DC gain and a low transient gain at the same time, a series compensation circuit can be used.

When a system exhibits negative damping characteristics, the voltage regulator usually aggravates the situation by increasing the negative damping. Great success has been achieved in reducing inter–area oscillations by using power system stabilizers to bring positive damping to the system. Supplementary signals are introduced through a compensation network to the excitation summation conjunction. This network provides the proper phase compensation so that the positive torque generated will be in phase with the change of speed of the machine.

Since the network transfer function changes when there is a change in system loading and/or network topology resulted from line switching or transmission fault, it is up to the designer to choose one situation where the parameters of the compensation network are tuned. In this section, the design of PSSs belonging to this category is reviewed.

There are many tuning strategies for determining the parameters of a conventional PSS. These tuning methods can be classified into the following categories, i. e., (1) classical control theory;(2) modern control theory; and (3) unclassified methodologies. The following review is presented according to this classification.

In 1975, Laha and Bollinger [1–3] proposed a PSS design procedure based on pole–placement. First, the frequency response is obtained by Fourier analysis of wide bandwidth excitation signals. Then, the transfer function of the generator is derived by fitting a straight–line approximation

to this frequency response. Finally the state–space model of the generator is obtained from this transfer function. In order to obtain the desired response, the closed–loop poles are placed using a PSS compensation network. The paper concludes that a PSS designed as such can damp the local oscillatory mode of the generator for the loading conditions at which the frequency response is obtained. In the same year, Bollinger et al [1–4] described a design method based on root locus techniques. In this algorithm, the open–loop poles and zeros of the generator are determined from the frequency response between the voltage regulator input and the electric power output of the generator. In [1–5], Bollinger and Lalonde designed a PSS compensation network to move a lightly–damped local mode to the left of the s–plane. To damp power oscillations on a tie–line between two large power pools, Bollinger, Winsor and Campbell [1–6] used a phase–compensation tuning technique to design a PSS based on frequency response data obtained by applying wide–bandwidth noise signals to the generator. In [1–7], Bollinger and Mistr designed a PSS for a pumped storage plant, using frequency response measurements and the root locus method.

There is a huge collection of papers on PSS design using eigenanalysis. De Mello et al [1–8] and others [1–9]~[1–13] have provided basic theory and practical applications of their techniques.

Modern control theory has been used to design optimal controllers. In 1972, Yu and Moussa proposed optimal PSS designs[1–14,1–15]. Later on

others proposed designs in the same category[1–16]~[1–18]. The problem with optimal control design is that the performance of the designed controller will deteriorate dramatically when the operating condition moves away from the point at which the controller was designed.

There are two basic papers on the application of conventional PSS. In 1974, de Mello and Concordia[1–2] presented the fundamental concepts of using supplementary excitation control to stabilize a synchronous generator. Using a one–machine infinite–bus system as the study system, the authors presented a complete design procedure to synthesize a PSS compensation network for generator stabilization. In 1981, Larsen and Swann presented three papers [1–19] on applying conventional power system stabilizers. These papers serve as both a survey and a tutorial.

1.2.3 Adaptive PSS Control

In the 1970s and early 1980s, adaptive control research progressed dramatically in two directions, i.e., self–tuning control and model reference control. At the same time, the availability of ever–increasing computing power made it possible to realize real–time on–line control, allowing relatively complex recursive identification algorithms to modify control actions in response to new system states.

The goal of adaptive control is to make the process under control less sensitive to changes in process parameters and the unmodeled process

dynamics. Adaptive controllers attempt to sense the changes of the dynamic process and make on-line adjustments to control parameters and strategies. There are two categories of adaptive control approaches, i.e., self-tuning control(STC) and model reference adaptive control(MRAC). With self-tuning control, system parameters are identified on-line using parameter identification schedules such as recursive least squares(RLS). Then the estimated parameters are utilized in the control strategy to modify control parameters. With model reference adaptive control, the output of the controlled process is compared with that of a reference model that exhibits the desired response. The error is used to modify controller parameters, with the objective of having the process output converge to the output of the reference model.

This is useful in control applications of power systems. With loads, generation schedules and network topology changing over time, the operating point of a power system changes. Fixed-parameter controllers designed for a specific operating condition can't maintain the same quality of performance at other operating points. There are a number of publications on this subject. A majority of the published works used self-tuning control schemes[1-20]~[1-26]. An MRAC scheme is used in [1-27]. They were mainly concerned with excitation control and load-frequency control.

1.2.4 Intelligent Control Systems

(1) *Promises of Artificial Intelligent Control*

Successful control of synchronous generators in a power system plays an important role in maintaining its continuous operation and integrity. Classical and modern control techniques using mathematical system theory, which deals with the analysis and synthesis of dynamical systems such as power systems, have performed remarkably well. However these classical techniques are inadequate when one is confronted with the control of complex systems characterized by poor models, high dimensionality of the decision space, multiple performance criteria, drifting parameter values, disturbances, failing component parts and nonlinearalities. These difficulties can be classified into three categories, namely, complexity in computation, nonlinearity with many degrees of freedom, and uncertainty. As the system to be controlled becomes more complex, so does the complexity associated with the computation of the control law and the task of implementing the control in a timely fashion. An intelligent control system is one that can sense the environmental changes, process the sensor data, generate and execute a timely control action that guides the system from an initial state to a terminal state satisfying the various constraints and objectives imposed. The greater the ability to successfully deal with the above said difficulties, the better the control system is.

Intelligent control, which is defined as a combination of control theory, operations research and artificial intelligence(AI) is emerging as one of the most popular new technologies in the industrial and manufacturing worlds. The assumption that all control engineering analysis and design can be reduced to the solution of a set of deterministic algebraic and differential equations has been challenged by these technologies. The data for plant modeling , measuring and controlling are rarely exact. Therefore the degree of uncertainty introduced in the modeling and controlling process becomes an important factor in influencing system performance. This is the driving force behind much of control engineering research.

Among the many possible new techniques based on AI, fuzzy logic is now the most visible and productive area. Since the first fuzzy chip in 1987, there have been 2000 patents issued in Japan. A majority of the patents issued in the US are from Japan on fuzzy logic controlled products, such as washing machines, air conditioners, camcorders, automobiles, subway trains and mobile robots. More recently, neural network theory has emerged as a complementary technique to fuzzy logic rather than competitive. Neural networks (a) are adaptive – they can learn from examples and then infer solutions from data presented to them; (b) can generalize – they can handle incomplete or imperfect data providing a measure of fault tolerance; (c) are non–linear and (d) are highly parallel. Motivated by the power of the human brain in parallel information processing, researchers have been exploring the

applicability of artificial neural networks(ANN) similar to a nervous system architecture to solve highly complex non–linear problems.

While fuzzy logic aims at exploiting the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost, neural networks mimic human beings to learn from examples. In contrast to fuzzy logic, where the knowledge is expressed explicitly, neural networks accommodate by coding it among the connecting weights of the network. Neural networks have no logical structure behind them.

In summary, fuzzy logic is good at various aspects of uncertainty–knowledge representation, but not powerful enough to carry out learning and adaptation to a changing environment, while neural networks are efficient structures capable of learning from examples, but they're purely numerical constructs. The learning of such a network is accomplished at the numerical level, and does not contribute towards developing new schemes of knowledge representation. Because of the uncertainty and complexity of the power system, the control of a synchronous generator is a good example where the combination of neural networks and fuzzy logic as a control system can challenge the classic design of control. The reason is that such an intelligent controller can provide robust control over the entire operating zone of the generator. In contrast, the performance of classic controllers deteriorates when saturation takes place or other nonlinearities come into play. There is

evident interest in combining both fuzzy logic and neural networks for power system control.

(2) *ANN Control*

Motivated by the power of the human brain in parallel information processing, researchers have been exploring the applicability of artificial neural networks (ANN) similar to nervous system architecture to solve highly complex nonlinear problems. It is reported[1–28] that generalized neural networks containing multilayer neural networks and linear dynamical systems as subsystems can be used to identify and to control a large class of nonlinear dynamic systems. Using such trainable networks as building blocks, one can formulate multilevel intelligent control systems for any dynamic system of varying complexity, thanks to the high capability of parallel information processing of the neural networks.

Unlike neural networks used in cognitive processing, those used in control systems are intentionally designed and trained in an environment in which the neural networks and the plant to be controlled are placed in a closed loop. That is, using neural networks to control a dynamic system requires that the neural networks be trained on–line. The specific training methods to be employed depends on the arrangement of the neurocontroller in the control loop. There are five generic approaches to designing a neurocontroller. They are supervised control including CMAC(Cerebellar

Model Arithmetic Computer), direct inverse control, neural adaptive control, backpropagation of utility, and adaptive critic.

In supervised control, a neural net learns the mapping from sensor inputs to desired actions by adapting to a training set of examples of what it should have done, i.e. it learns to imitate a human or a computer program which already knows how to perform the task. The CMAC approach seems to hold promise in on–line learning where classical backpropagation shows poor performance because of its slow learning. This method has found applications in function approximation, pattern recognition and robotic control[1–29].

In direct inverse control, a neural net learns the inverse dynamics of the plant to be controlled. That is, the network receives the desired system trajectory as an input and produces the correct control action. A major problem with inverse control arises when many plant inputs produce the same output, i.e. when the plant's inverse is not well defined. In this case, the neural network will attempt to map the same input to many different target responses. In both supervised control and direct inverse control, there is no planning or optimization. In neural adaptive control, linear mappings used in standard designs such as MRAC(Model Reference Adaptive Control) are replaced by neural nets, resulting in greater robustness and ability to handle nonlinearity.

The backpropagation of utility adapts an optimal controller essentially by solving a calculus of variations problems. That is to develop a schedule of

actions so as to maximize utility over a period of time. The emphasis is placed on performance over a period of time rather than instantaneous performance. As with the calculus of variations, this method requires a model of the system to be controlled (which may itself be a neural net) and does not allow for noise.

The adaptive critic approach can be thought of as either a generalization of backpropagation of utility or a generalization of reinforcement learning. It does not require a model of the system. The underlying idea is to approximate the Bellman equation of dynamic programming. Reinforcement learning is a very general approach to learning that can be applied when the knowledge required to use supervised learning is not available. It basically involves two problems. The first one is to construct a critic capable of evaluating plant performance in a way that is both appropriate to the actual control objective and informative enough to allow learning. The second problem is to determine how to alter controller outputs to improve performance as measured by the critic.

In the past decade, the field of neural networks has grown from a mere handful of researchers to thousands spread throughout the academic and industrial world. Applications have been developed in domains as varied as character recognition, pattern classification, dynamic system control, and stock market prediction. Specifically, as applied to power system engineering[1–30], neural networks have found their applications in areas

such as nuclear power generation, system operation, modeling and prediction, control, load forecasting, scheduling and optimization, and fault diagnosis. Many of these problems have solutions in normal logic and analytical methods. With the use of neural networks(NNs), there seems to be simpler and faster methods of solving these problems. On the other hand, there seems to have been a big gap between research and implementation of these research results in utilities, especially those related to safety systems.

(3) *FL Control*

Since the first paper on fuzzy sets was published in 1965 by Lotfi A. Zadeh[1–31], applications of fuzzy theory have penetrated not only in the fields of soft sciences such as biological and social sciences, linguistics, psychology and economics, but also in those fields where dependencies of variables are well–defined. This latter category of applications is advantageous as precision is illusory in the real world. Fuzzy algorithms possess a remarkable innate ability to exploit the tolerance for imprecision to achieve tractability, robustness and low cost solution. In contrast, traditional control techniques fail to do so when using precise algorithms to arrive at a solution.

Fuzzy control has found successful applications in industrial processes such as the Automatic Train Operation(ATO) at Sandai, as well as electronic consumer products such as washing machines, vacuum cleaners. Today, we are experiencing the turning point, i.e., a paradigm shift from traditional hard computing to soft computing. The latter is mainly involved with fuzzy

logic(FL), artificial neural network theory(ANN), and probabilistic reasoning(PR). FL is concerned with imprecision and approximate reasoning; ANN with learning, generalization and error tolerance; and PR with uncertainty and propagation of belief.

One important characteristic of an intelligent control system is that it has the ability to control a system in uncertainty or unknown environments. Fuzzy set theory provides a systematic framework for dealing with different types of uncertainty within a single conceptual framework. This is why fuzzy inference procedures are crucial to the process of managing uncertainty.

(4) *Literature Review*

The application of neural networks to the control of synchronous machines is at a very early stage. The control problem may concern any of the following three subsystems, i.e., the prime-mover frequency control, excitation terminal voltage control and low frequency oscillation control which involves coupling among machines connected through transmission lines. Several control schemes in this area have been published.

A nonlinear adaptive controller based on neural networks has been proposed very recently for control of synchronous generators[1-32]. The control system consists of an identifier and a controller. The connection weights of the neural networks are adjusted adaptively so as to respond to changes of operating conditions. Digital simulations of the control scheme have been carried out for a one-machine infinite bus system. In [1-33], an

artificial neural network (ANN), functioning as a power system stabilizer (PSS), has been used to map the inverse dynamics of the controlled plant, a one–machine infinite bus system. During training, the ANN is presented with input–output pairs of the plant with an adaptive PSS tuned in. In operation, the ANN is fed with the desired output of the synchronous machine and the output of the ANN is used as the control signal to the controlled plant. In [1–34] an adaptive forced–action automatic excitation controller for a salient–pole synchronous generator has been proposed based on an ANN which is trained using input–output patterns. The backpropagation algorithm was chosen for training the ANN.

A neurocontroller for load frequency control has been proposed in [1–35] in which a feedforward neural network is trained to control the steam admission value of the turbine that drives the generator, thereby restoring the frequency to its nominal value. Different from conventional integral control which is slow, the proposed neurocontroller uses an estimator (which may itself be neural net) to account for load perturbation in driving the plant controller. In [1–36], on the basis of a cluster–wise segmented associative memory system, real–time adaptation of a neurocontroller for a power system is carried out using an error–based on–line learning scheme. The capability of self–organization and predictive estimation of neural net computers has been utilized in the design. In [1–37], an on–line trained neurocontroller has been proposed for turbogenerators based on a hierarchical architecture of

neural networks. A nonlinear adaptive controller based on neural networks has been proposed very recently for control of a synchronous generator[1-38]. The control system consists of an identifier, a controller and a reference model. The connection weights of the neural networks are adaptively adjusted so as to respond to changes of operating conditions. Digital simulations of the control scheme have been carried out for a multi-machine power system.

Other subjects that have drawn much attention are DC motor and induction motor controls due to their simpler dynamics. Reported methodologies can be broadly divided into two categories; one is to model the existing controller by an ANN and replace that controller with the trained neurocontroller; the other to identify the dynamics of the motor system on-line using an ANN and then construct a control strategy using the updated ANN weights.

Several intelligent control schemes have been proposed in the last few years, based either on neural networks or fuzzy logic for either voltage control or system stability control. Their hardware implementations have not been widely reported. Only has Mitsubishi filed its patent in the US for a fuzzy logic power system stabilizer(PSS). In this section, representative papers are reviewed. A PID type fuzzy logic PSS is proposed by Hiyama[1-39,1-40]. The controller uses real power of the generator as input. After integrating the input signal twice, acceleration, speed and phase signals are

derived which are used as the inputs to a fuzzy logic controller. The output is the control signal to the excitation loop which is supposed to provide a damping torque in phase with the change of speed. There are three to five parameters to be tuned on site if such a PSS is to be installed. Tuning parameters of the design on site is always a disadvantage, just as for conventional PSSs.

An automatic voltage regulator is designed for a synchronous generator, based on fuzzy logic control[1–41]. Its task is to maintain the terminal voltage of the generator constant. It is not concerned with transient stability. A self–organizing fuzzy controller is proposed to improve transient stability of a multimachine system through additive real power control which may be implemented by a braking resistor, fast valving or superconducting magnetic storage unit[1–42]. The controller is equivalent to an expert system to evaluate the right amount of accelerating energy to be shed in order to stabilize the generator. The potential of the excitation system is not exhausted as the control signal does not go to this channel, unless there is another power system stabilizer.

1.3 Statement of Objectives

The goal of this research is to develop an intelligent controller that is robust to changing environment and inexpensive to implement. The control of a synchronous generator in a power system is different from controlling most

other industrial processes in that the dynamic of one generator is affected by that of all other generators, loads, topologies of the transmission network as well as types of disturbances. It would be too expensive, too complicated and impractical to have all this information fed back to the controller to arrive at an optimum control strategy.

For example, to improve transient stability through excitation control, the key point is to provide a damping torque in phase with the change of speed to the generator. In practice, this is achieved by using a conventional power system stabilizer(CPSS) which is nothing but a phase lead–lag compensation circuit. When the operating point of the system changes, the parameter–fixed compensation circuit may not provide adequate phase compensation anymore. Therefore, the controller may not perform as desired. This is why researchers have been looking for better designs based on AI techniques.

Intelligent controllers will be designed based on a technique that combines the purely numerical approach of neural networks and the structured–knowledge representation of fuzzy logic. The functions of the neural networks will be limited to state estimation and pattern recognition. The architecture of the entire control system remains fuzzy. Specifically, this thesis aims at (1)investigating the possibilities of using advanced control theory for power system stability control, (2)designing advanced intelligent control systems for synchronous generator control in a multi–machine

environment, (3)developing an advanced simulation environment for power system transient stability studies. The following tasks were carried out to achieve the stated objectives.

- Review of literature pertaining to power system stability and control using conventional and adaptive control theory; presentation of a combined theory of neural networks and fuzzy logic.
- Improvement of the Transient Stability Simulation Package(TSSP)[1–43] implemented in SIMULINK environment. Addition of new features and functions for extended transient stability studies.
- Design and performance studies of an ANN–FL(artificial neural network and fuzzy logic) controller as applied to the speed–tracking control of a DC motor.
- Design and performance evaluation of two ANN–FL PSSs for a multi–machine power system. Simulated results are compared with those obtained using a conventional PSS.

1.4 Outline of the Thesis

The research reported in this thesis has been focused on developing advanced intelligent control systems for stability control of power system synchronous generators.

A review of stability and control problems in a large power system is conducted in Chapter 1. Due to the interconnection of local power systems

into large power pools, stability problems appeared. Negative damping introduced by high gain fast–acting AVRs and weak inter–ties between pools resulted in lightly damped oscillations.

Artificial neural networks (ANNs) and fuzzy–logic(FL) can be used to model complex dynamic systems. Basic concepts, definitions of related terms used in ANNs and FL are given in Chapters 2 and 3. A feedforward ANN architecture and the four types of fuzzy reasoning methods are described. Combining the merits of ANNs and FL, an hybrid modeling methodology is proposed in Chapter 4.

Utilizing the hybrid modeling technique, Chapter 5 designs an intelligent controller for speed–tracking control of a DC motor. Comparison of transient responses of the system under different disturbances are performed using a PID controller and the newly–designed ANN–FL controller. Conclusions are drawn from the simulation studies.

Modeling aspects of composite power systems are dealt with in Chapter 6. Mathematical representations of various power components, including synchronous generators, induction motors, high voltage AC/DC transmission networks, are given. Block diagrams of excitation control systems, speed–governing and turbine systems as well as power system stabilizers are presented.

A computer software for transient stability simulation studies of a power system is developed in Chapter 7 in SIMULINK environment. The

Transient Stability Simulation Package (TSSP) [1–43] is realized using basic SIMULINK blocks and MATLAB functions. Basic instructions on how to build a simulation model of a power system are given.

The hybrid ANN–FL modeling technique is applied to a two–machine infinite–bus power system in Chapter 8. Based on experience gained in Chapter 5, two ANN–FL power system stabilizers, one using speed deviation as input, the other using accelerating power, are designed. Their performance evaluation is carried out by simulation studies.

Chapter 9 concludes this thesis.

CHAPTER 2

ARTIFICIAL NEURAL NETWORKS

This chapter presents some basic concepts of artificial neural networks(ANNs), starting with a biological neuron, and its modeling by an artificial neuron. Three basic activation or transfer functions are then described. A multi–layer feedforward neural network can be formed by organizing a large number of single artificial neurons in layers. Such a network can be trained for use in system modeling and control.

2.1 INTRODUCTION

Modern computers can perform a variety of well–defined tasks such as inversion of large matrices or solving differential equation systems at speeds unmatched by humans. On the other hand, man–made machines can't solve many other problems to our satisfaction yet, even though these problems can be easily disentangled by the cognitive power of humans. The brain is more effective than the digital computer when it comes to solving problems that are ill–defined and that need massive processing. To recognize a face in a crowd, the human brain uses a gigantic web of interconnected processing elements called neurons to process information from sense organs. The human brain accomplishes massive parallel information processing with

millions and even billions of neurons in parts of the brain working together. The vast processing power inherent in biological neural structures has inspired the study of the structure itself for hints on organizing man–made computer structures. Artificial neural networks(ANNs) are an attempt to imitate the biological paradigm, our brain, in structure and in function. Their creation is aimed to technically realize capabilities and characteristics such as self–organization, learning and associative memory. To achieve this, a large number of simple processor elements(PEs) are interconnected with uni–directional signal channels to single– or multi–layer networks. All these PEs work in parallel for sequential arithmetic and/or symbolic information processing. As solid state operational amplifiers and logic gates can perform computations many times faster than the biological neurons in the brain[2–1,2–2], we may expect to build relatively inexpensive machines with the capability to process as much information as the human brain.

Definition 2.1: An artificial neural network is a massively connected processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects[2–1]; (1) knowledge is acquired by the network through a learning process; and (2) interneuron connection strengths known as synaptic weights are used to store the knowledge.

The primary similarity between biological nervous systems and artificial neural networks is that each of the two consists of a large number of

simple elements that can learn and are able to collectively solve complicated and ambiguous problems[2–3]. Today, most research on ANNs is accomplished by simulating these networks on serial computers. Speed limitations have kept such networks relatively small. However, their applications are versatile. Apart from those reviewed in Chapter 1, more general applications include truck control system[2–4], robot control[2–5], and medical diagnosis[2–6]. Other excellent references are listed as [2–7]~[2–10].

In this chapter, the fundamental working principle of a biological neuron and its modeling by an artificial neuron as a processing element(PE) is presented. When a large number of such PEs are organized topologically to form a network, it will possess the ability of learning, adaptation and generalization.

2.2 A Biological Neuron

Neural networks are aggregates of interconnected nerve cells. A nerve cell comprises the cell body(soma) which surrounds the cell nucleus. The cell body has a long processus, i.e., the axon which ends in numerous ramifications attached to other cells via so–called synaptic end–heads thus forming the synaptic connections between neurons[2–2]. A branching structure called dentite picks up signals from other neurons. At the tail end of the axon, these are brushlike structures called synaptic buttons, as shown in Figure 2.1.

In the stationary electrochemical state the cell has a resting potential of about –80mV. If a cell is stimulated by another cell via the synaptic connection, a pole reversal of about 30mV results in a very short period of time. This action or activation potential moves across the axon to neighboring cells with a speed of up to 100 meters per second. To generate this action potential, the stimulus must exceed a specific threshold. The activation potential drops immediately after its rise and the cell returns to the original state, as shown in Figure 2.2. The reaction of a neuron to a stimulus received depends on its history, for instance, how many impulses it has transmitted before and in which sequence. This is the learning and adaptive capability of the biological neural networks, or the plasticity characteristic of the synapses which gives the neurons a memory.

2.3 MATHEMATICAL MODEL OF A NEURON

On the working principle of the biological neuron, a mathematical model can be constructed as shown in Figure 2.3, where θ_k is the threshold value, h_k the inhibitory input. The neuron acts like an integrator with feedback. The integrator integrates the weighted presynaptic input signals and maps the postsynaptic activation x_j on to the output signal by means of a nonlinear function. The differential equation for the activation u_k describes the short–term behavior of the neuron as a function of the input signal x_i , i.e.,

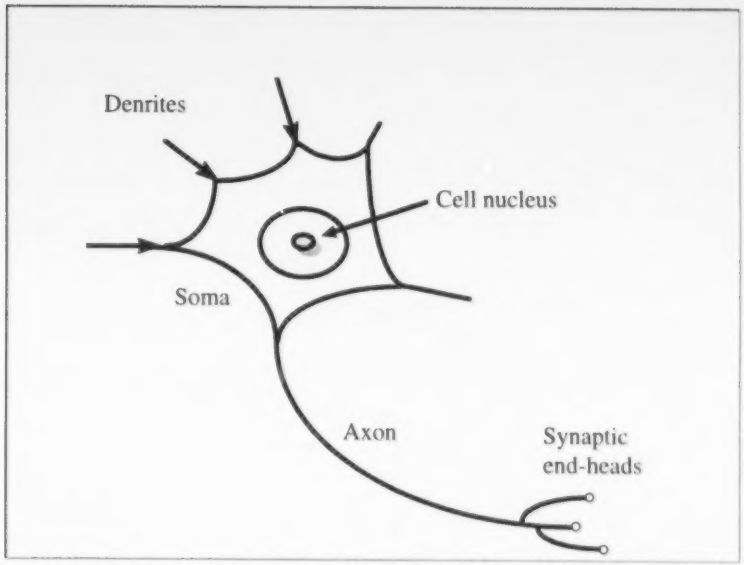


Fig. 2.1 A Biological Neuron

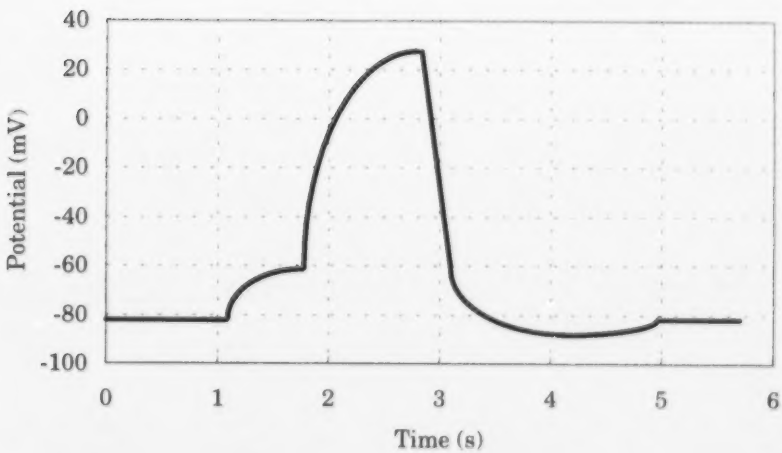


Fig. 2.2 Response to A Stimulus

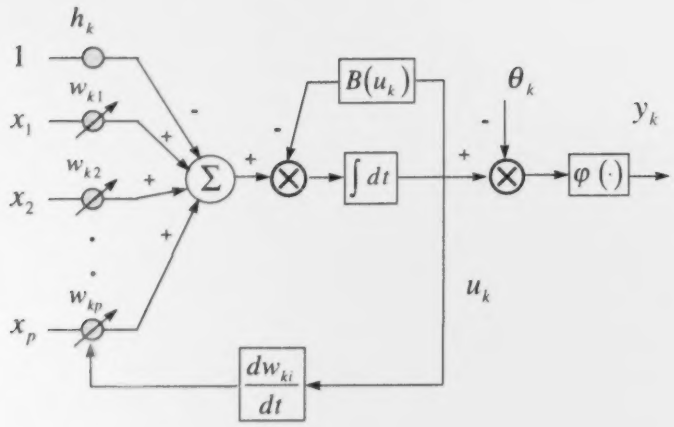


Fig. 2.3 Mathematical Model of the Neuron

$$\frac{du_k}{dt} = -B(u_k) + \sum_{i=1}^{n} w_{ki} x_i - h_k \tag{2.1}$$

To model the plasticity property of the biological neuron, the following differential equation is used to adapt the connecting weights, i.e.,

$$\frac{dw_{ki}}{dt} = \eta(t) \cdot \varphi(x_i, u_k, w_{ki}) \tag{2.2}$$

It describes the dynamics of learning as a function of the instantaneous values of the connecting weights, the activation and the input quality. $\eta(t)$ is a learning factor governing the adapting speed of the weights.

A further simplified model can be obtained as shown in Figure 2.4. It was the earliest contribution using such a model to simulate the way neural nets work. The output of the model can be expressed as

$$y_k = \varphi(u_k - \theta_k) = \varphi\left(\sum_{j=1}^{p} w_{kj} x_j - \theta_k\right) \tag{2.3}$$

where

x_i = the input signals,

w_{ki} = the synaptic weights of neuron k;

u_k = the output of the linear combiner;

θ_k = the threshold;

φ = an activation function; and

y_k = the output signal of the neuron.

The threshold θ_k can be both positive and negative. When it is positive, it has the effect of lowering the net input of the activation function. When it is negative, it has the effect of increasing the net input of the activation function. In the latter case, the threshold becomes a bias which is the negative of the threshold. Replacing θ_k in (2.3) by a weight w_k with a constant input x_0 fixed at either -1 or $+1$, corresponding to θ_k being negative or positive, respectively, the final model of the neuron is given by Figure 2.5. Biological neurons are living cells capable of receiving and transmitting electrochemical signals in highly specialized ways. Their complexities can be accurately simulated only by intricate computer chips. The modeling of the biological neurons by artificial neural nets are extremely simplified representations of the real thing.

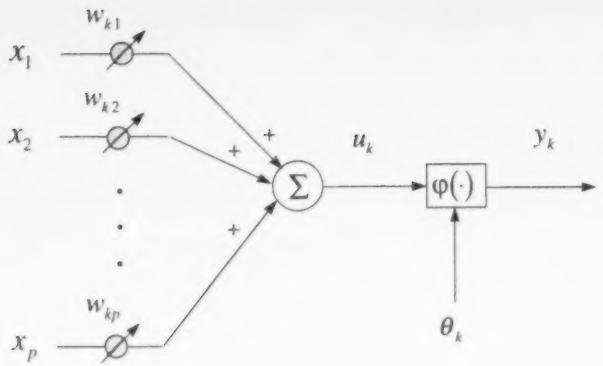


Fig. 2.4 Nonlinear Model of A Neuron

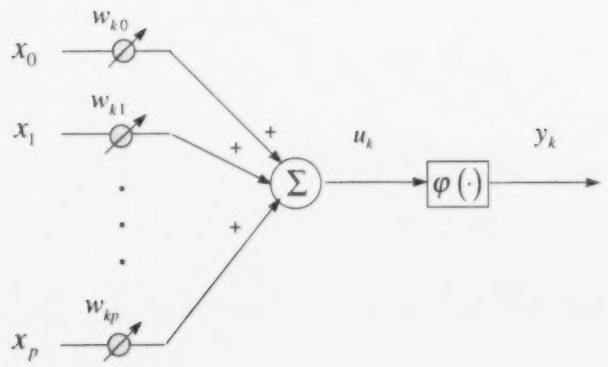


Fig. 2.5 Final Model of the Neuron

2.4 ACTUATION FUNCTIONS

To describe the output, y, of a neuron in terms of the net activity level, x, numerous nonlinear transfer functions can be utilized. We present three of the most commonly used transfer functions here. For a more comprehensive listing and description of these and other functions, please refer to[2-1,2-10].

(1) *Symmetric hard Limit Transfer Function*

Shown in Figure 2.6, this transfer function forces a neuron to produce an output of +1 if its net input reaches a threshold, otherwise a −1. It is often used in a network for decision making or classification. The mathematical expression is given by

$$y = \begin{cases} +1, & x \geq 0, \\ -1, & x < 0. \end{cases} \tag{2.4}$$

(2) *Tan Sigmoid Transfer Function*

Shown in Figure 2.7, this transfer function provides saturation for decision making, yet they have differentiable input-output characteristics that facilitate adaptivity. It maps the input of a neuron from the interval $(-\infty,+\infty)$ to the interval $(-1,+1)$. The mathematical expression is given by

$$y = \tanh\left(\frac{1-e^{-2x}}{1+e^{-2x}}\right) \tag{2.5}$$

(3) *Log Sigmoid Transfer Function*

Shown in Figure 2.8, this squashing transfer function can be viewed as performing a function similar to an analog electronic amplifier. The gain of the amplifier is analogous to the ratio of the change in output for a given change in input. When the inputs are near zero, the gain is greatest. This will mitigate problems associated with possible dominating effects of large input signals. The mathematical expression of the transfer function is given as follows:

$$y = \frac{1}{1+e^{-x}} \tag{2.6}$$

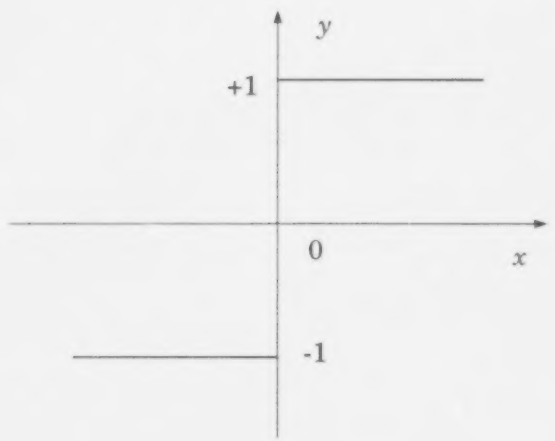


Fig. 2.6 Symmetric Hard Limit Activation Function

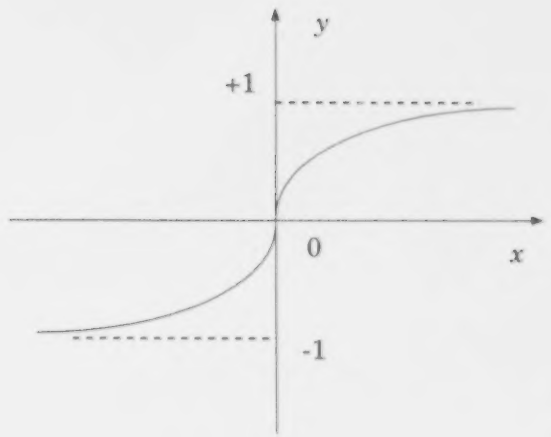


Fig. 2.7 Tanh Sigmoid Activation Function

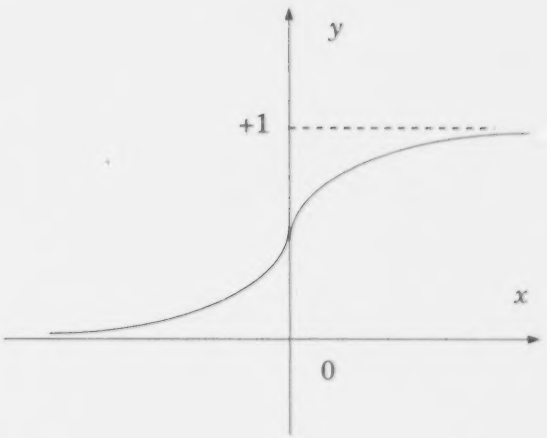


Fig. 2.8 Log Sigmoid Activation Function

2.5 NEURAL NETWORK ARCHITECTURES

An artificial neuron described by the above model is extremely simple. For solving complex problems, a large number of such simple processing elements(PEs) can be organized topologically to form different structures of networks. The combined characteristic features of parallel processing, adaptive connection weights, and nonlinear output functions yield the possibility for the design of neural networks with unprecedented computing power. There are different ways to categorize network architectures. Depending on the mathematical equations used to describe the model, two types of architectures can be identified, i.e., one by nonlinear differential equations and the other by algebraic expressions. According to the actual structure of the network, four classes of network architectures can be identified[2–1]. They are:

- single–layer feedforward networks;
- multilayer feedforward networks;
- recurrent networks and
- lattice structures.

In this thesis, only multilayer feedforward networks are used. Therefore, the structure of such networks is discussed in more detail. A layered neural network is one whose neurons are organized in the form of layers. Figure 2.9 shows a typical multilayer feedforward network. Because the signals flow

only in one direction, i.e., from left to right, the network is called feedforward. On the leftmost is an input layer. On the rightmost is an output layer. The layers in between are called hidden layers. The neurons in hidden layers are called hidden neurons or hidden units. This network is said to be fully connected as every node in each layer is connected to every other node in the adjacent forward layer. If some of these connections are missing, such a network is called a partially connected network.

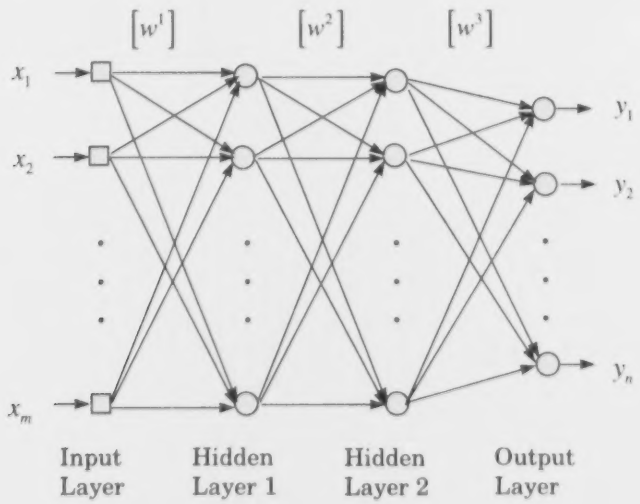


Fig. 2.9 A Feedforward Multilayer Neural Network

2.6 LEARNING RULES

In designing a neural network for modeling a process or a plant, one important step is to teach the network the environment. The network improves its performance through learning. The realization of this learning is

achieved through an iterative process of adjustments of the synaptic weights and thresholds. The learning process is complicated and hard to define with precision.

Definition 2.2: Learning is a process by which the free parameters of a neural network are adaptive through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameters adjustment takes place.

There are many learning rules such as error–correction learning rooted from optimum filtering, Heblian learning and competitive learning inspired by neurobiological considerations, and Boltsman learning borrowed from thermodynamics and information theory. On the other hand, a model of the environment in which the designed neural network operates is referred to as a learning paradigm. There are three learning paradigms, i.e., supervised learning, reinforcement learning by use of critic, and self–organized learning which is also called unsupervised learning where no teacher or critic is required. In this thesis, supervised learning is used. This subject will be further covered in Chapter 4.

2.7 MULTILAYER PERCEPTRONS

In this section, the detailed derivation of the delta training rule for multilayer perceptrons, i.e., multilayer feedforward neural networks is presented. Such networks can be successfully trained using the supervised

paradigm called back–propagation algorithm based on error–correction learning.

The training process of the algorithm consists of two passes of a signal through the network. In the forward pass, input data are applied to the input layer as stimuli and are propagated to the output neurons where a set of outputs are obtained and compared with the teacher's output values. An error vector is then produced. In the backward pass, the error vector is backward propagated to adjust the synaptic weights. This process continues until the norm of the error vector is brought to a satisfactory small number. From Figure 2.10, the error of output neuron j at iteration k can be expressed as

$$e_j^k = d_j^k - y_j^k \tag{2.7}$$

where d_j^k is the desired output of neuron j at iteration k.

The total squared error of the output layer is given by

$$E^k = \frac{1}{2}\sum_{j=1}^{N_o}\left(e_j^k\right)^2 \tag{2.8}$$

where N_o is the total number of neurons at the output layer.

If there are N_p patterns of training data, then the average squared error

$$E = \frac{1}{N_p}\sum_{k=1}^{N_p} E^k = \frac{1}{2N_p}\sum_{k=1}^{N_p}\sum_{j=1}^{N_o}\left(e_j^k\right)^2 \tag{2.9}$$

The goal when training a neural network is to minimize E by adjusting the synaptic weights. In Figure 2.10, the net internal activation of neuron j at iteration k, v_j^k, is given by

$$v_j^k = \sum_{i=0}^{N_m} w_{ji}^k y_i^k \tag{2.10}$$

where

N_m = the number of neurons at the $m-$th hidden layer,

$w_{j0} = \theta_j$, the threshold; and

$y_0 = -1$.

Therefore, the output of output neuron j at iteration k is given by

$$y_j^k = \varphi\left(v_j^k\right) \tag{2.11}$$

Now our task is to find out how to modify w_{ji}, knowing everything else. Using the chain rule in calculus, we have

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial e_j}\frac{\partial e_j}{\partial y_j}\frac{\partial y_j}{\partial v_j}\frac{\partial v_j}{\partial w_{ji}} \tag{2.12}$$

where the superscript k in the formula is omitted.

Carrying out the derivatives, we can have

$$\frac{\partial E^k}{\partial w_{ji}^k} = -e_j^k \cdot \varphi'\left(v_j^k\right) \cdot y_j^k \tag{2.13}$$

The correction of w_{ji}^k can then be made by

$$w_{ji}^{k+1} = w_{ji}^{k} + \Delta w_{ji}^{k} \tag{2.14}$$

where

$\Delta w_{ji}^{k} = -\eta \cdot \dfrac{\partial E^{k}}{\partial w_{ji}^{k}}$, called delta rule, and

η = a learning rate that controls the learning speed.

For a hidden neuron j, the above formula has to be modified. Please refer to [2–1] for details.

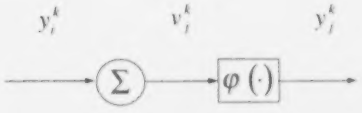


Fig. 2.10 Output Neuron j

CHAPTER 3

FUZZY SETS AND FUZZY REASONING

This chapter introduces some fundamental concepts of fuzzy sets, fuzzy operations and approximate reasoning. It gives the definition of linguistic variable, membership function, fuzzy relation, fuzzy logic and approximate reasoning. Four fuzzy inference and two defuzzification methods are described. These concepts and definitions will be used in the subsequent chapter to present a neural fuzzy–logic modeling technique.

3.1 INTRODUCTION

Motivated by the conviction that traditional methods of systems analysis are unsuited for dealing with systems in which relations between variables do not lend themselves to representation in terms of differential or difference equations, Lotfi A. Zadeh proposed the first paper on fuzzy sets in 1965[3–1]. During the past several years, applications of fuzzy set theory have become productive and fruitful in industrial process control, because of the very active research in this area[3–2]. In short, fuzzy set theory is a body of concepts and techniques that give a form of mathematical precision to human reasoning which is imprecise and ambiguous. Contrary to traditional

numerical techniques, much of human reasoning involves the use of linguistic variables whose values are fuzzy sets rather than exact numerical numbers.

Zadeh's principle of incompatibility states that as the complexity of a system to be modeled increases, our ability to make precise and yet significant statements about the system's behavior, using traditional system modeling techniques, diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics[3–3, 3–4].

3.2 FUZZY SETS

In crisp sets, the transition of an element between membership and nonmembership in a given set is abrupt. The membership value is either 1 or 0. In fuzzy sets, this transition can be gradual. The boundaries between fuzzy sets are vague and ambiguous. The membership value of an element in a fuzzy set is measured by a function that attempts to describe vagueness and ambiguity.

Let U be a collection of objects denoted by $\{u\}$. U is called the universe of discourse and u represents a generic element of U.

Definition 3.1: A fuzzy set $\underline{X}$ in a universe of discourse U is characterized by a membership function u_x which takes values in the interval [0,1], namely, $\mu_x : U \rightarrow [0, 1]$.

Fuzzy sets are denoted by a set symbol with a understrike. A fuzzy set $\underline{X}$ in U may be represented as a set of ordered pairs of a generic element u and its grade of membership function, i.e.,

$$\underline{X} = \{(u, \mu_x(u)) | u \in U\} \tag{3.1}$$

where $\mu_x(u) \in [0, 1]$. When U is continuous and infinite, the fuzzy set $\underline{X}$ can be written as

$$\underline{X} = \int \frac{\mu_x}{u} \tag{3.2}$$

When U is discrete and finite, the fuzzy set $\underline{X}$ can be written as

$$\underline{X} = \frac{\sum_{i=1}^{n} \mu_x(u_i)}{u_i} \tag{3.3}$$

In both notations, the horizontal bar is not a quotient, but rather a delimiter. The numerator is the membership value associated with the element of the universe indicated in the denominator. The "Σ" sign denotes a fuzzy union and the integral sign denotes a set union for continuous variables.

3.3 Fuzzy Set Operations

Let $\underline{A}$ and $\underline{B}$ be two fuzzy sets in U with membership functions μ_A and μ_B, respectively. The theoretic set operations of union, intersection and complement for fuzzy sets are defined via their membership functions.

Union: $$\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\} \tag{3.4}$$

Intersection: $\mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$ (3.5)

Complement: $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$ (3.6)

The diagrams in Figure 3.1 show these operations.

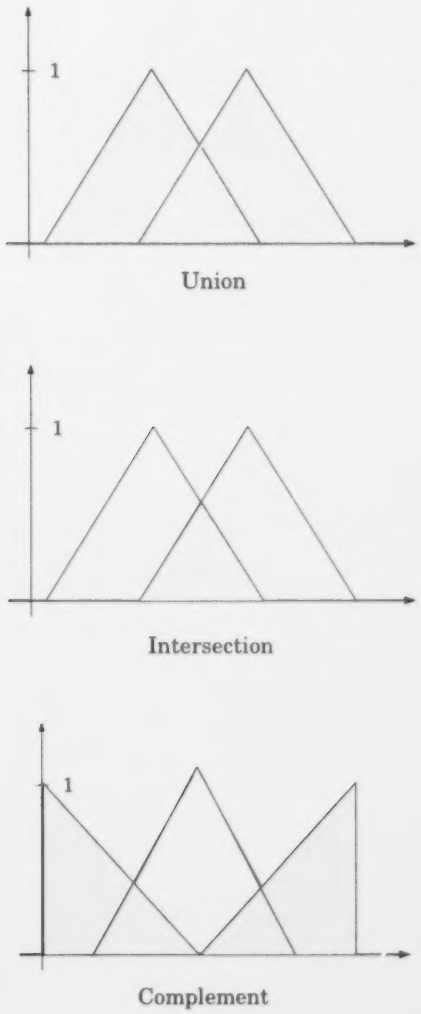


Fig. 3.1 Fuzzy Set Operations

If $A_1, A_2, \ldots, A_n$ are fuzzy sets(the understrike is omitted for simplicity), in $U_1, U_2, \ldots, U_n$, respectively, the Cartesian product of $A_1, A_2, \ldots, A_n$ is a fuzzy set in the product space $U_1 \times U_2 \times \cdots \times U_n$ with the membership function

$$\begin{aligned}\mu_{A_1 \times A_2 \times \cdots \times A_n}(u_1, u_2, \ldots, u_n) &= \min\left\{\mu_{A_1}(u_1), \mu_{A_2}(u_2), \ldots, \mu_{A_n}(u_n)\right\} \\ &= \mu_{A_1}(u_1) \times \mu_{A_2}(u_2) \times \ldots \times \mu_{A_n}(u_n)\end{aligned} \tag{3.7}$$

To express ambiguous relationships such as "you and your brother look very similar", and "x and y are almost equal", we use fuzzy relations instead of crisp relations.

Definition 3.2: The fuzzy relation R between V and U is a fuzzy set in the Cartesian Product expressed as

$$R_{V \times U} = \left\{\left((V, U), \mu_R(v, u)\right) \middle| (v, u) \in V \times U\right\} \tag{3.8}$$

The fuzzy relation in $U_1 \times U_2 \times \cdots \times U_n$ is given by

$$\begin{aligned}R_{U_1 \times U_2 \times \cdots \times U_n} = \Big\{&\left((u_1, u_2, \ldots, u_n), \mu_R(u_1, u_2, \ldots, u_n)\right) \\ &\Big| (u_1, u_2, \ldots, u_n) \in U_1 \times U_2 \times \cdots \times U_n\Big\}\end{aligned} \tag{3.9}$$

Other concepts used in this thesis are presented as follows.

Definition 3.3: Sup–star Composition: If R and S are fuzzy relations in $U \times V$ and $V \times W$, respectively, the composition of R and S is a fuzzy relation denoted by $R \bullet S$ and is defined by

$$R \bullet S = \left\{\left[(u, w), \sup_V\left(\mu_R(u, v) * \mu_S(v, w)\right)\right], u \in U, v \in V, w \in W\right\} \tag{3.10}$$

where $*$ could be any operator, namely, minimum, algebraic product, bounded product, or drastic product defined as follows.

Triangular norm: The triangular norm is a two–place function from [0,1] $[0,1]\times[0,1]\rightarrow[0,1]$, which includes the following operations defined for all $x,y\in[0,1]$:

Intersection: $x\wedge y=\min\{x,y\}$ (3.11)

Algebraic: $x\cdot y=xy$ (3.12)

Bounded: $x\otimes y=\max\{0,x+y-1\}$ (3.13)

Drastic Product: $$x\cap y=\begin{cases}x,y=1;\\ y,x=1;\\ 0,x,y<1.\end{cases}\tag{3.14}$$

The *support* of a fuzzy set $\underline{X}$ is the crisp set of all points u in U such that $\mu_{\underline{x}}(u)>0$. The element u in U at which $\mu_{\underline{x}}=0.5$, is called the *crossover* point. A fuzzy set whose support is a single point in U with $\mu_{\underline{x}}=1$, is referred to as a fuzzy *singleton*. The *core* of a membership function for a fuzzy set $\underline{X}$ is defined as the region in U where $\mu_{\underline{x}}(u)=1$. The boundaries of a membership function are the region in U where $0<\mu_{\underline{x}}(u)<1$. Figure 3.2 illustrates the various regions as defined above. A normal fuzzy set $\underline{X}$ is one whose membership value is unity at least at one point in U. A convex fuzzy set $\underline{X}$ is one whose membership function satisfies

$$\mu_{\underline{X}}\left(\lambda u_1+(1-\lambda)\mu_2\right)\geq\min\left\{\mu_{\underline{X}}(u_1),\mu_{\underline{X}}(u_2)\right\}\tag{3.15}$$

where $u_1,u_2\in U$ and $\lambda\in[0,1]$.

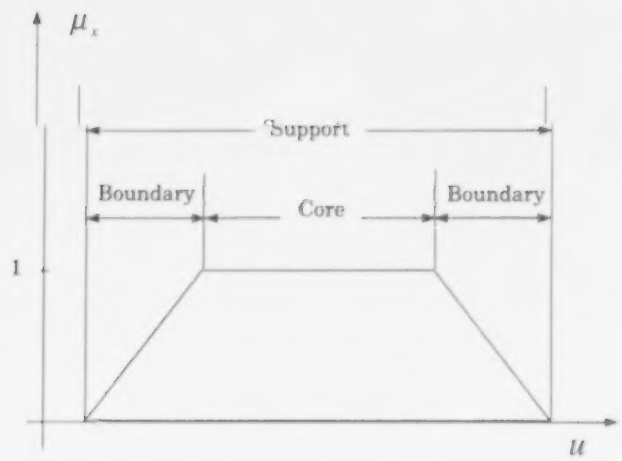


Fig. 3.2 Support, Core and Boundary

3.4 LINGUISTIC VARIABLES

A linguistic variable can be regarded as a variable whose value is a fuzzy number or as a variable whose values are defined in linguistic terms. A linguistic variable is characterized by a quintude $(u, T(u), U, G, M)$ in which u is the name of the variable; $T(u)$ is the term set of u, that is, the set of names of linguistic values of u with each value being a fuzzy number defined on U; G is a syntactic rule for generating the names of values of u; and M is a semantic rule for associating with each value its meaning. For example, if temperature is interpreted as a linguistic variable, then its term set $T(temp)$ could be

$$T(temp) = \{low, medium, high\} \tag{3.16}$$

where each term in *T(temp)* is characterized by a fuzzy set in a universe of discourse $U = [10,80]$. We might interpret '*low*' as 'a temperature below about 20', and '*medium*' as 'a temperature near 50', and '*high*' as 'a temperature above about 80'. These terms can be characterized as fuzzy sets whose membership functions are shown in Figure 3.3.

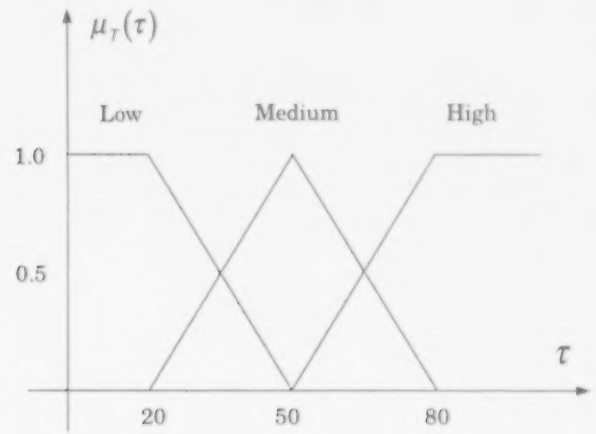


Fig. 3.3 Membership Functions of Temperature

3.5 Fuzzy Logic and Approximate Reasoning

Logic deals with propositions that may or may not be true. A proposition can be true on one occasion, but false on another. That 'A person of 5'7" is tall' is such an example. If this person is seen walking down an Edmonton street, this proposition may not hold true anymore. The truth value of the statement can be only one or zero, depending on whether it is true or false, respectively. By assuming only a true or false value to such a statement, we use crisp

logic. On the other hand, the proposition in the example can be said to be true to a certain degree, say 0.8, instead of 1. This is equivalent to saying that he is tall to the degree of 0.8. Here we are using fuzzy logic. Propositions that take truth values to a certain degree – from 0 to 1 are dealt with by fuzzy logic. A fuzzy logic proposition is a statement without clearly defined boundaries. The truth value assigned to a proposition can be any value on the interval [0,1]. The assignment of the truth value to a proposition is actually a mapping of the truth value from the interval [0,1] to the universe U. Propositions can be combined to form new propositions by means of logical operations.

Approximate reasoning or fuzzy inference plays a central role in all human thinking activities as well in the inquiry of the meaning of various soft or empirical sciences. Instead, certain form of approximate and vague reasoning is always involved apart from statistical inference in analysis of data. Unlike conventional expert systems where statements are executed consequential, the principal reasoning protocol behind fuzzy logic is a parallel processing paradigm where all the rules are fired. Each rule is a statement of relationship between model variables and fuzzy regions. When these rules are evaluated for their degrees of truth, they contribute to the output state of the solution variable set. The functional tie between the degree of truth in related fuzzy regions is called the method of implication. The functional tie

between the fuzzy regions and the expected value of a set point is called the method of defuzzification.

3.6 INFERENCE METHODS

Fuzzy logic control describes algorithms for process control as a fuzzy relation between the state condition of the process to be controlled and the input to the process. A control algorithm comprises a set of fuzzy control rules expressed by "IF – THEN" expressions. The "IF" clause of the rules is called the *antecedent* and the "THEN" clause the *consequent*.

State variables have fuzzy values expressed by fuzzy sets. The output of a fuzzy control system is calculated by means of fuzzy inference given the state variables. The rule base of a fuzzy logic control(FLC) is usually defined from expert knowledge. For a multi–input single–output(MISO) system, the rule base has the form

$$R = \{RB_1, RB_2, \ldots, RB_g\} \tag{3.17}$$

where RB_j represents the rule:

IF (x_1 is A_{1j} and x_2 is A_{2j} ... x_m is A_{mj}) THEN u is B_j

where

x_i = linguistic variables representing the process states,

u = control variable,

A_{ij}, B_j = linguistic labels $(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$ of the above linguistic

variables in the universe of discourse $V_1, V_2, \ldots, V_m$ and U, respectively.

Using the triangular norms, we can define conjunctions in approximate reasoning. A fuzzy control rule, "if x is $\underline{X}$ then y is $\underline{Y}$," is represented by a fuzzy implication function and is denoted by $\underline{X} \rightarrow \underline{Y}$, where $\underline{X}$ and $\underline{Y}$ are fuzzy sets in universes U and V with membership functions $\mu_{\underline{X}}$ and $\mu_{\underline{Y}}$, respectively.

The fuzzy conjunction is defined by

$$\underline{X} \rightarrow \underline{Y} = \underline{X} \times \underline{Y}$$

$$= \int_{U \times V} \mu_{\underline{X}}(u) * \mu_{\underline{Y}}(v) / (u, v), \text{ all } u \in U \text{ and } v \in V \tag{3.18}$$

where $*$ is an operator representing a triangular norm.

To represent different types of fuzzy reasoning, we define two fuzzy implication functions. By using the definition of the fuzzy conjunction, Mamdani's mini-fuzzy implication, R_c, is obtained if the intersection operator is used, i.e.,

$$R_c = \underline{X} \times \underline{Y}$$

$$= \int_{U \times V} \mu_{\underline{X}}(u) \wedge \mu_{\underline{Y}}(v) / (u, v) \tag{3.19}$$

This is the mini-operation rule of fuzzy implication by Mamdani[3-5]. Similarly, Larsen's product fuzzy implication, R_p, is obtained if the algebraic product is used, i.e.,

$$R_p = \underline{X} \times \underline{Y}$$

$$= \int_{U \times V} \mu_x(u) \cdot \mu_y(v) / (u,v) \tag{3.20}$$

This is the production operation rule of fuzzy implication by Larsen[3–6]. Now we are able to present the four types of fuzzy reasoning currently utilized in FLC applications.

(a) *Type 1 Fuzzy Reasoning*

This type of reasoning is associated with the use of Mamdani's minimum operation rule, R_c, as a fuzzy implication function. Suppose that the system is a multi–input single–output (MISO) system, the membership function of the control action produced by the j–th rule is given by

$$\mu_{B_j} = \alpha_j \wedge \mu_{B_j}(u) \tag{3.21}$$

where α_j is the firing strength of the j–th rule and is expressed as

$$\alpha_j = \mu_{A_{1j}}\left(x_1^0\right) \wedge \mu_{A_{2j}}\left(x_2^0\right) \wedge \ldots \wedge \mu_{A_{mj}}\left(x_m^0\right) \tag{3.22}$$

with $x_i^0 (i = 1,2,\ldots,m)$ being a fuzzy singleton and $\wedge$ denoting intersection. The membership function of the total control action of the n–rules can be obtained by

$$\mu_B(u) = \mu_{B_1} \vee \mu_{B_2} \vee \ldots \vee \mu_{B_n} \tag{3.23}$$

with $\vee$ denoting union. This fuzzy reasoning process is graphically illustrated in Figure 3.4. Assuming that $x_1 = x_1^0$ and $x_2 = x_2^0$. From fig. 3.4, it can be determined that

$$\alpha_1 = \mu_{A_{11}}(x_1^0) \wedge \mu_{A_{21}}(x_2^0) = \mu_{A_{21}}(x_2^0) \tag{3.24}$$

$$\alpha_2 = \mu_{A_{12}}(x_1^0) \wedge \mu_{A_{22}}(x_2^0) = \mu_{A_{12}}(x_1^0) \tag{3.25}$$

The value of the control variable produced by the first rule equals the trapezoid area in μ_{B_1}. The value of the control variable produced by the second rule equals the trapezoid area in μ_{B_2}. The final control action is produced by centroid defuzzification from the two trapezoids(see Section 3.7).

(b) *Type 2 Fuzzy Reasoning*

This type of reasoning is associated with the use of Larsen's product operation rule, R_p, as a fuzzy implication function. In this case, the membership function of the control action produced by the j–th rule is given by

$$\mu_{B_j} = \alpha_j \cdot \mu_{B_j}(u) \tag{3.26}$$

The membership function of the total control action of the n–rules can be obtained by the same formula as (3.22). Figure 3.5 shows this fuzzy reasoning process. The final control action is produced through defuzzification(see Section 3.7).

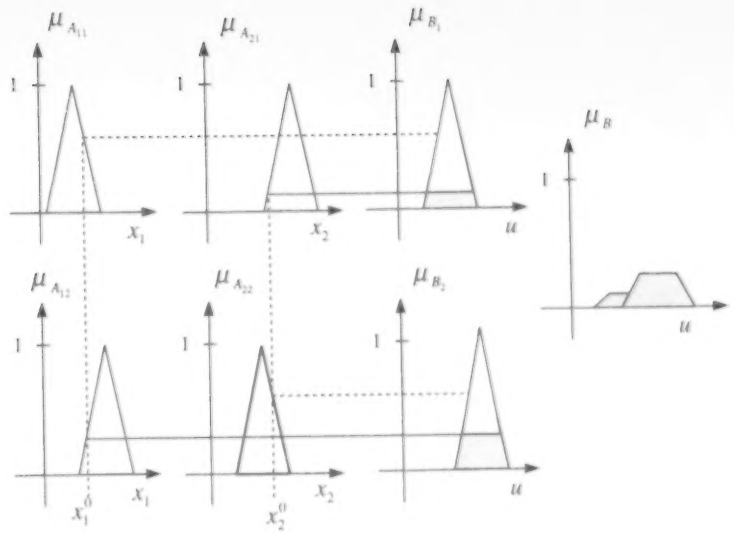


Fig. 3.4 Type 1 Fuzzy Reasoning

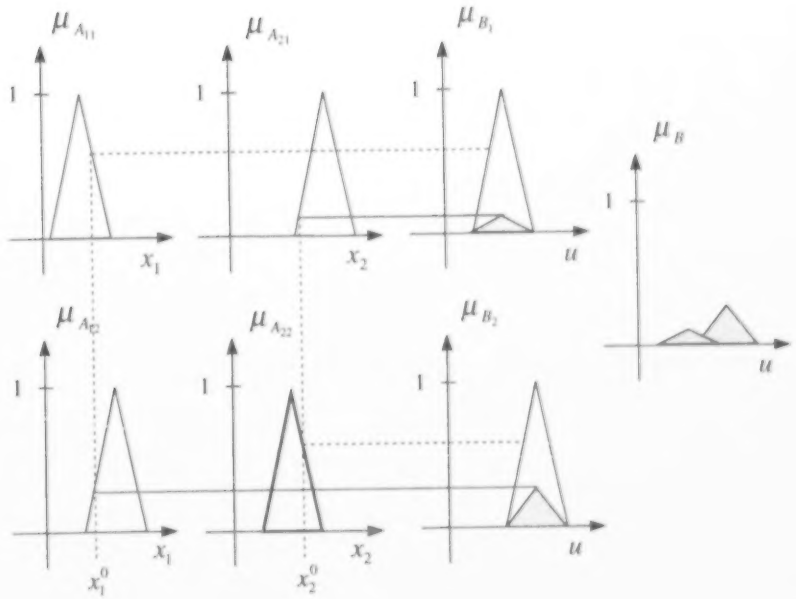


Fig. 3.5 Type 2 Fuzzy Reasoning

(c) *Type 3 Fuzzy Reasoning*

It was proposed by Tsukamoto[3–7]. In this mode of reasoning, the membership functions of fuzzy sets $A_{ij}(i=1,2,...,m;\ j=1,2,...,n)$ are single valued, and $B_j(j=12,...,n)$ are monotonic. The firing strength for the j–th rule is also given by (3.22). From this α_j, the output variable u_j can be determined by letting $\alpha_j=\mu_{B_j}(u_j)$ when μ_{B_j} is monotonic. A crisp control action may be expressed as the weighted combination of all the outputs, i.e.,

$$u_o=\frac{\sum_{j=1}^{n}\alpha_j u_j}{\sum_{j=1}^{n}\alpha_j} \tag{3.27}$$

(d) *Type 4 Fuzzy Reasoning*

It uses a function of the input linguistic variables as the consequence of a rule. The j–th fuzzy control rule is of the form

IF (x_1 is A_{1j} and x_2 is A_{2j} ... x_m is A_{mj}) THEN $u=f_j(x_1,x_2,...,x_m)$

where f_j is a function of the process variables $x_i(i=1,2,...,m)$ defined in the input subspace. The inferred value of the control action from the j–th rule is $\alpha_j f_j(x_1^0,x_2^0,...,x_m^0)$, and the total crisp control action is given by

$$u_o=\frac{\sum_{j=1}^{n}\alpha_j f_j(x_1^0,x_2^0,...,x_m^0)}{\sum_{j=1}^{n}\alpha_j} \tag{3.28}$$

3.7 DEFUZZIFICATION

In the previous Section, we have outlined four fuzzy reasoning methods. In the first two methods, a space of fuzzy control actions is produced instead of a space of a crisp control actions. However, in practical applications, the latter is required. Therefore, to map from a fuzzy space to a crisp space, defuzzication is required.

Defuzzification is the final stage of fuzzy reasoning[3–8,3–9]. The evaluation of the fuzzy model propositions is done through an aggregation process such that a final fuzzy region for each solution variable is produced. This region is then decomposed using one of the defuzzification strategies.

(1) *Composite Maximum*

A maximum decomposition finds the domain point with the maximum truth. This technique applies to a narrower class of problems. Its attributes are that (i) the expected value is sensitive to a single rule that dominates the fuzzy rule set and (ii) the expected value tends to change from one point to another as the shape of the fuzzy region changes. In applications that need to assess the maximum fuzzy property, such as risk assessment, this may be the technique to use.

(2) *Composite Moments(Centroid)*

The centroid or center of gravity generates the center of gravity of the possibility distribution of a control action. Mathematically, in the case of a continuous universe, this method produces

$$u_o = \frac{\int \mu_{B'}(u) u du}{\int \mu_{B'}(u) du} \tag{3.29}$$

where $\int$ denotes conventional integral.

We have presented the two most commonly used methods of defuzzification. Unless we have reason to believe that a specialized method of defuzzification is required for our model, we will use one of these two techniques.

CHAPTER 4

HYBRID ANN-FUZZY LOGIC MODELING

This chapter presents a hybrid modeling methodology, using artificial neural networks(ANNs) and fuzzy–logic(FL). ANNs are unstructured numerical estimators that can learn, generalize and adapt. Fuzzy–logic is a structured modeling system. Two architectures of ANN–FL models are discussed. Detailed learning rules for Type 4 fuzzy reasoning are derived using the chain rule. A backpropagation algorithm is chosen to be the backbone for developing the related formulas.

4.1 INTRODUCTION

In the previous two chapters, we have presented two model-free dynamic systems, i.e., neural networks and fuzzy logic inference systems. Both have the capability of dealing with difficulties arising from uncertainty, imprecision and noise. Fuzzy logic control systems are structured modeling systems. They enjoy broad applications in various fields[4–1]. The design of fuzzy logic control systems has been mainly carried out by trial-and-error methods[4–2,4–3], based on expert knowledge of the system operation. The fuzzy sets of input and output variables, the membership functions can all be modified individually or collectively, if the design fails. This modification is

usually subjective and manual. It is a time-consuming process. This task may become even more difficult in designing MIMO systems. To circumvent these difficulties, self-organizing FL controller and parameter estimation design have been proposed[4–4,4–5].

Neural networks are unstructured trainable numerical estimators. They are characterized with the capabilities of learning, generalization, fault tolerance and high efficient parallel information processing[4–6]. By integrating the low-level learning and computation power of ANNs with the high-level thinking of fuzzy logic systems, we can automate the design of such intelligent control systems. Active investigations in this area have been reported in [4–7]~[4–11]. A fuzzy neural net is one that possesses fuzzy signals and/or fuzzy weights. In [4–12], three types of fuzzy neural nets are discussed. They are fuzzy neural networks with crisp input signals but fuzzy weights; fuzzy input signals but crisp weights; and fuzzy input signals and fuzzy weights, respectively.

4.2 FUZZY LOGIC CONTROL SYSTEM

To effectively model a real world system that is imprecise and uncertain, we use fuzzy logic instead of the traditional logic systems. Using a set of linguistic control rules and a fuzzy inference system, a fuzzy logic control system can deal with source information that is suited only for the human

decision–making process. Essentially, it consists of five functional blocks as shown in Figure 4.1.

(1) *Fuzzification*:

It transforms the state variables which are crisp into fuzzy sets in the input universe of discourse. In a control application, the observed data are usually crisp. Since variable manipulation in a FLC is based on fuzzy set theory, fuzzification becomes a first necessary step. One practical procedure for fuzzification is to convert a crisp value into a fuzzy singleton. More specifically, fuzzification includes the following steps: (a) measuring the values of the input variables; (b) scaling the ranges of the input variables to their corresponding universes of discourse; and (c) finding the linguistic values of the input variables

(2) *Rule Base* :

It consists of a set of linguistic rules in the form: IF A THEN B, where A and B are labels of fuzzy sets characterized by appropriate membership functions. This set of rules is established on expert knowledge of the dynamic behavior of the fuzzy system under study. Fuzzy control rules provide a convenient way for expressing control policy and domain knowledge.

(3) *Data Base*:

It defines membership functions of the fuzzy sets used in the rule base of an application. Many techniques are involved in forming the data base.

Experience and engineering judgement will also be used in designing the data base of a fuzzy logic control.

(4) *Decision-Making Logic*:

It imitates human decision-making based on fuzzy concepts and infers fuzzy control actions by employing inference operations on the rule base.

(5) *Defuzzification*:

It transforms the inferred fuzzy results into a crisp output. Using the general rules of fuzzy inference as covered in chapter 3, the evaluation of a single proposition yields one fuzzy set. The aggregation of all the fuzzy sets produced from this evaluation process forms a fuzzy region of the solution variable. Defuzzification is a process which finds the best scalar representation of the information contained in this solution fuzzy set.

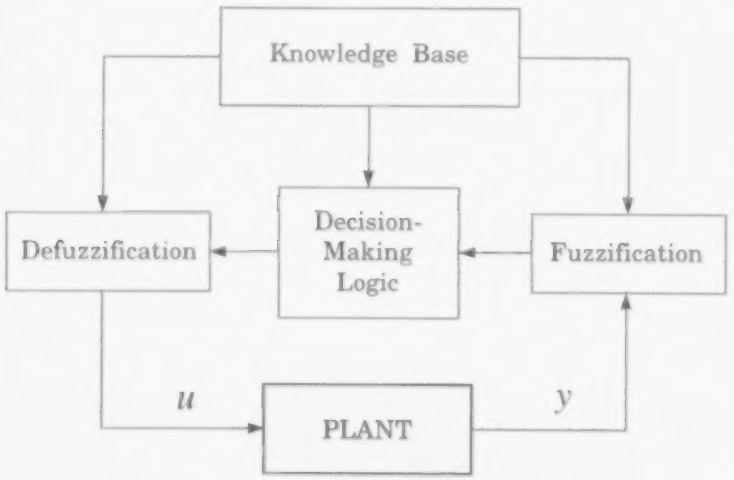


Fig. 4.1 Fuzzy Logic Control System

4.3 ARCHITECTURE OF ANN-FUZZY LOGIC MODELING

A nonlinear dynamic system can be modeled by an artificial neural network(ANN) to an arbitrary accuracy as long as there are enough neurons in the network. In this case the state and solution variables are all crisp numbers. What if a system can only be described by fuzzy variables or a combination of fuzzy and crisp variables? Fuzzy neural nets can be used to model such systems. For a neural net to be called a fuzzy neural net, the signals and/or the weights must be fuzzy sets.

An ANN-fuzzy logic model is a multilayer feedforward network with each 'neuron' performing a specified function on incoming signals and connecting weights. According to the use of different inference methods, we present two architectures of ANN-fuzzy logic models. In Figure 4.2, the model consists of five layers. In Figure 4.3, the model consists of four layers.

Layer 1: The outputs of this layer are $\mu_{A_{ij}}(x_i)$, where x_i are the inputs, A_{ij} $(i = 1,2,...,m; j = 1,2,...,n)$ are the linguistic labels. The membership functions $\mu_{A_{ij}}$ specify the degree to which the given x_i satisfy A_{ij}. For example, we can use the bell-shape membership function

$$\mu_{A_{ij}}(x_i) = \exp\left\{-\left(\frac{x_i - m_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{4.1}$$

where m_{ij}, σ_{ij} are the parameters for the membership function $\mu_{A_{ij}}$. They will be determined in the learning process.

Layer 2: Node j in this layer defines the firing strength of rule j in the rule base according to the input $x_i (i = 1,2,...,m)$, i.e.,

$$\alpha_j = \mu_{A_{1j}}(x_1^0) \wedge \mu_{A_{2j}}(x_2^0) \wedge ... \wedge \mu_{A_{mj}}(x_m^0) \tag{4.2}$$

Layer 3: In Figure 4.2, depending on whether Type 1 or Type 2 reasoning is used, each node performs a product $(\cdot)$ or a minimization $(\wedge)$ operation. The output of node j in this layer can be expressed as

$$\mu_{B_j} = \alpha_j * \mu_{B_j}(u) \tag{4.3}$$

where $*$ denotes either product $(\cdot)$ or intersection $(\wedge)$. In Figure 4.3, Type 3 or Type 4 reasoning is used. Each node performs a product $(\cdot)$ operation. The output of node j in this layer can be expressed as

$$\mu_{B_j} = \alpha_j \cdot g_j \tag{4.4}$$

where

$$g_j = \begin{cases} \mu_{B_j}^{-1}(\alpha_j), \textit{Type 3;} \\ f_j(x_1^0, x_2^0, ..., x_m^0), \textit{Type 4.} \end{cases} \tag{4.5}$$

Layer 4: In this layer, aggregation is carried out for both models.

Layer 5: Defuzzification produces a crisp output in each of these two models. Note that the integral sign "$\int$ " indicates fuzzy union.

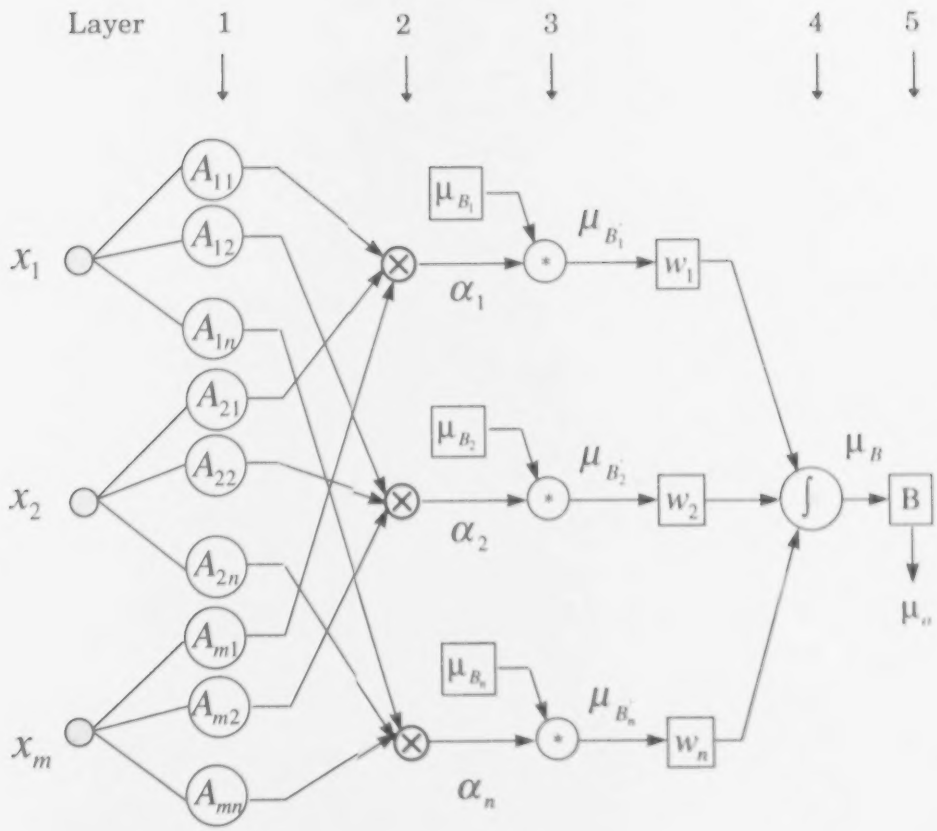


Fig. 4.2 ANN-FL Modeling, Types 1 & 2 Reasoning

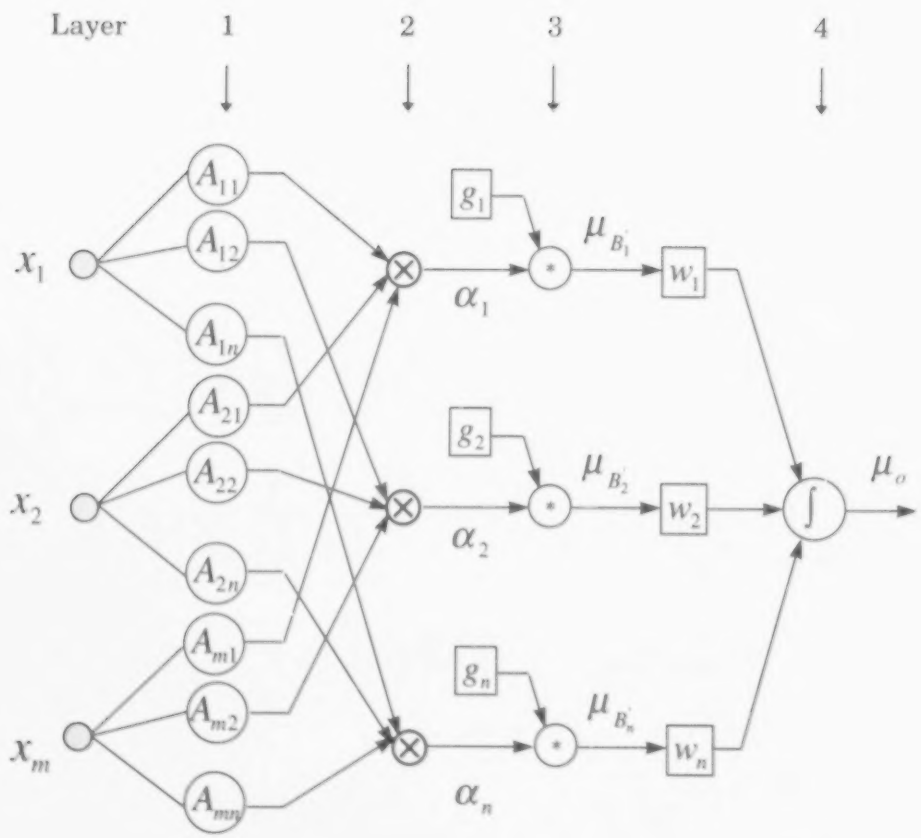


Fig. 4.3 ANN-FL Modeling, Types 3 & 4 Reasoning. For Type 4 reasoning, the Input Vector x Has to Be Fed to the Function Vector g

In the following section, using Type 4 reasoning as an example, we present details on net input, activation function and net output of each node from layer 1 to layer 4. Assuming that the net input to node k of layer l is given by

$$x_k^l = \sum_{j=1}^{N_{l-1}} w_{kj}^{l-1} \cdot y_j^{l-1} \tag{4.6}$$

where $N_{l-1}(l = 1,2,...,4)$ is the number of nodes of the previous layer who have direct connections with present node, w_{kj}^{l-1} the connection weights, y_j^{l-1} the net outputs of the nodes of the previous layer given by

$$y_k^{l-1} = f_k^{l-1}\left(x_k^{l-1}\right) \tag{4.7}$$

where f_k^{l-1} are the activation functions of the nodes.

Layer 1: The net input, output, and activation function for layer 1 are identified as follows,

$$x_k^l = x_i \tag{4.8}$$

$$y_k^l = \mu_{A_{ij}}\left(x_k^l\right) \tag{4.9}$$

$$f_k^l(u) = \mu_{A_{ij}}(u) \tag{4.10}$$

where $l = 1, i = 1,2,..,m; j = 1,2,...,n; k = 1,2,...,(m \times n)$.

Layer 2: The net input, output, and activation function for layer 2 are identified as follows,

$$x_k^l = \min_i\left(y_{[(i-2)\times n+k]}^{l-1}\right) \tag{4.11}$$

$$y_k^l = x_k^l = \alpha_k \tag{4.12}$$

$$f_k^l(u) = u \tag{4.13}$$

where $l = 2, i = 1,2,..,m; k = 1,2,...,n$.

The synaptic weights from layer 1 to layer 2 are unity.

Layer 3: The net input, output, and activation function for layer 3 are identified as follows,

$$x_k^l = \alpha_k \cdot g_k = \mu_{B_j} \tag{4.14}$$

$$y_k^l = x_k^l \tag{4.15}$$

$$f_k^l(u) = u \tag{4.16}$$

where $l = 3, k = 1,2,..,n, w_k^l = 1$; and $g_k = \left(x^0, a_j\right)$ in which the state variable is expressed as $x = \left[x_1, x_2, ..., x_m\right]^T$ and the parameter vectors of the membership functions as $a_j = \left[a_{1j}, a_{2j}, ..., a_{mj}\right]^T$.

Layer 4: The net input, output, and activation function for layer 4 are identified as follows,

$$x_k^l = \sum_{j=1}^{n} w_j \cdot \mu_{B_j} \tag{4.17}$$

$$y_k^l = \frac{x_k^l}{\sum_{j=1}^{n} \alpha_j} \tag{4.18}$$

$$f_k^l(u) = \frac{1}{\sum_{j=1}^{n} \alpha_j}(u) \tag{4.19}$$

where $l = 4, k = 1,2,..,n; w_k^l = w_k$.

4.4 LEARNING RULES

According to the nature of the dynamic system under study, expert knowledge is used to determine the rule base and the universe of discourse of the fuzzy variables in the system. Once this is established, the basic structure of the ANN-FLC model will be formed. Now our task is to optimally determine the parameters of the membership functions. This is similar to the determination of the synaptic weights in conventional neural networks. Applying the delta-rule as outlined in Chapter 2 and the backpropagation algorithm, we will derive the formula for determining the adjustable parameters in the model.

The problem of optimally adjusting these parameters can be stated as follows: given the training data $x_i(t)\left(i = 1,2,...,m; t = 1,2,...,N_p\right)$, the desired output value $u_j(t)\left(j = 1,2,...,N_o; t = 1,2,...,N_p\right)$, the structure of the ANN-FLC model, and the rule base, determine the synaptic weights of the model such that the overall squared error between the actual output of the model and the desired output is minimum, where N_o is the number of output neurons, N_p the number of training patterns.

Suppose that there are L layers and the l-th layer has N_l nodes. Denote the i-th node of the l-th layer by (l,i) and its node output by y_i^l. As the output of a node is a function of its incoming signals, we can write the node output as

$$y_i^l = f_i^l\left(y_1^{l-1}, y_2^{l-1}, \ldots, y_{N_{l-1}}^{l-1}; a_i\right) \tag{4.20}$$

where $y_i^{l-1}\left(i = 1,2,\ldots,N_{l-1}\right)$ are the outputs of the previous layer and a_i is a vector of parameters of the activation function at node (l,i).

In order to identify a measure of how well a network is performing on the training data, we define an error term that depends on the difference between the output value of an output neuron referred as the target value t_j, and the value it actually has as a result of the feedforward calculations, i.e., $y_i^L\left(i = 1,2,\ldots,N_o\right)$. This error term is defined for each given pattern(there are N_p patterns) and summed over all output neurons for that pattern. Specifically, for output neuron j and pattern p, we define

$$e_{pj} = t_{pj} - y_{pj}^L \tag{4.21}$$

The total squared error of the output layer is given by

$$E_p = \frac{1}{2}\sum_{j=1}^{N_o}\left(t_{pj} - y_{pj}^L\right)^2 \tag{4.22}$$

The total average squared error of the output layer over all patterns of the training set is given by

$$E = \frac{1}{2N_p}\sum_{j=1}^{N_p} E_p \tag{4.23}$$

The goal of the training process is to minimize the average sum squared error over all training patterns. Assuming that w, the synaptic weight, is the

adjustable parameter in a node (i.e., parameter of a membership function), the general learning rule can be expressed as

$$w^{k+1} = w^k + \eta \cdot \left(-\frac{\partial E}{\partial w} \right)_k \tag{4.24}$$

where η is the learning rate and k denotes iteration. Using the chain rule, we can show how to calculate $\frac{\partial E}{\partial w}$ in the formula and the error to be propagated for each layer of the ANN-FLC system as illustrated in Figure 4.3.

Layer 4: The error to be propagated to the previous layer is given by

$$\frac{\partial E}{\partial y_{pj}^L} = -\left(t_{pj} - y_{pj}^L \right) \tag{4.25}$$

For convenience in calculating the derivatives of the error function, we restate the chain rule here.

$$\frac{\partial E_p}{\partial y_{pj}^l} = \sum_{k=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_{pk}^{l+1}} \frac{\partial y_{pk}^{l+1}}{\partial y_{pj}^l} \tag{4.26}$$

Applying this rule, we have the formula for calculating the derivative of the error function with respect to the connection weights and parameters of the membership functions of the consequence part of each rule in the rule base.

$$\frac{\partial E_p}{\partial w_j^l} = \frac{\partial E_p}{\partial f_k^l} \frac{\partial f_k^l}{\partial x_k^l} \frac{\partial x_k^l}{\partial w_j}$$

$$= -\left(t_{pk} - y_{pk}^L \right) \frac{1}{\sum_{j=1}^{n} \alpha_j} \alpha_j \cdot g_j \left(x^0, a_j \right) \tag{4.27}$$

$$\frac{\partial E_p}{\partial a_{kj}^l} = \frac{\partial E_p}{\partial f_k^l}\frac{\partial f_k^l}{\partial x_k^l}\frac{\partial x_k^l}{\partial a_{kj}}$$

$$= -\left(t_{pk} - y_{pk}^L\right)\frac{1}{\sum_{j=1}^{n}\alpha_j}\sum_{i=1}^{n} w_i \alpha_i \cdot \frac{\partial g_i}{\partial a_{kj}} \tag{4.28}$$

where $j = 1,2,...,n; k = 1,2,...,m+1$. Using (4.24), we can update the weights w_j and a_j (both are vectors) as the presentation of pattern p is completed. The error to be propagated backward is given by

$$\delta_j^L = t_{pj} - y_{pj}^L \tag{4.29}$$

Layer 3: As the weights are unity and no membership function parameters to be adjusted, only the error signal needs to be computed.

$$-\delta_j^l = \frac{\partial E_p}{\partial y_{pj}^l}$$

$$= \sum_{k=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_{pk}^4}\frac{\partial y_{pk}^4}{\partial y_{pj}^l}$$

$$= \sum_{k=1}^{N_4} -\left(t_{pk} - y_{pk}^L\right)\frac{w_j}{\sum_{i=1}^{n}\alpha_i}$$

$$= -\left(t_p - y_p^L\right)\frac{w_j}{\sum_{i=1}^{n}\alpha_i} \tag{4.30}$$

$$\delta_j^l = \left(t_p - y_p^L\right)\frac{w_j^{l+1}}{\sum_{i=1}^{n}\alpha_i} \tag{4.31}$$

where $l = 3, L = 4$, and $j = 1,2,...,n$.

Layer 2: This layer is similar to layer 3. We calculate the error signal only.

$$-\delta_j^l = \frac{\partial E_p}{\partial y_{pj}^l}$$

$$= \sum_{k=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_{pk}^3} \frac{\partial y_{pk}^3}{\partial y_{pj}^l}$$

$$= \sum_{k=1}^{N_3} -\left(t_p - y_p^L\right) \frac{w_k^4}{\sum_{i=1}^{n} \alpha_i} \frac{\partial y_{pk}^3}{\partial y_{pj}^l}$$

$$= -\left(t_p - y_p^L\right) \frac{w_j^4}{\sum_{i=1}^{n} \alpha_i} \cdot g_j \text{, only when } k = j. \tag{4.32}$$

$$\delta_j^l = \left(t_p - y_p^L\right) \frac{w_j^4}{\sum_{i=1}^{n} \alpha_i} \cdot g_j \tag{4.33}$$

where $l = 2, L = 4$, and $j = 1,2,\ldots,n$.

Layer 1: This is the most complicated layer when it comes to calculating the derivatives. In this layer, there are $m \times n$ nodes, each of which represents a membership function with undetermined parameters. It is in the learning process that those parameters are adjusted.

$$\frac{\partial E_p}{\partial y_{pj}^l} = \sum_{k=1}^{N_{l+1}} \frac{\partial E_p}{\partial y_{pk}^2} \frac{\partial y_{pk}^2}{\partial y_{pj}^l}$$

$$= \sum_{k=1}^{N_2} -\left(t_p - y_p^L\right) \frac{w_k^4}{\sum_{i=1}^{n} \alpha_i} \cdot g_k \cdot \frac{\partial y_{pk}^2}{\partial y_{pj}^l} \tag{4.34}$$

$$\frac{\partial y_{pk}^2}{\partial y_{pj}^1} = \begin{cases} 1, \text{ if } y_{pk}^2 = y_{pj}^1 = \min_i \{\mu_{A_{ik}}\}, \\ 0, \text{ otherwise.} \end{cases} \tag{4.35}$$

Therefore, the error function is given by

$$\delta_j^l = \sum_{k=1}^{N_2} -(t_p - y_p^L) \frac{w_k^4}{\sum_{i=1}^{n} \alpha_i} \cdot g_k \tag{4.36}$$

This formula holds only if $y_{pk}^2 = y_{pj}^1 = \min_i \{\mu_{A_{ik}}\}$, otherwise the error is zero.

$$\frac{\partial E_p}{\partial m_{rk}} = \frac{\partial E_p}{\partial y_{pk}^l} \frac{\partial y_{pk}^l}{\partial m_{rk}}$$

$$= \delta_k^l \cdot \mu_{A_{rk}} \frac{2(x_r - m_{rk})}{\sigma_{rk}^2} \tag{4.37}$$

$$\frac{\partial E_p}{\partial \sigma_{rk}} = \frac{\partial E_p}{\partial y_{pk}^l} \frac{\partial y_{pk}^l}{\partial \sigma_{rk}}$$

$$= \delta_k^l \cdot \mu_{A_{rk}} \frac{2(x_r - m_{rk})^2}{\sigma_{rk}^3}, \text{ where } l = 1. \tag{4.38}$$

CHAPTER 5

ANN–FLC APPLICATION: SPEED–TRACKING CONTROL OF A DC MOTOR

On the basis of the hybrid artificial neural network and fuzzy logic (ANN–FL) modeling presented in Chapter 4, this chapter walks through a complete design process of an ANN–FL controller for a DC motor system. Results from simulation studies of the newly–designed controller are presented. It is shown that the ANN–FL controller has fast response when the system experiences a load or reference speed change. Also the ANN–FL control system provides perfect speed–tracking control.

5.1 INTRODUCTION

The hybrid ANN–FL modeling technique has been presented in Chapter 4. As an example of application of this method to a simple dynamic system, the dynamics of a direct–current motor will be analyzed and an adaptive controller using the hybrid theory designed. Performance studies of the designed controller will be carried out and a comparison with a PID controller performed. This design procedure will lay a foundation for carrying out stability and control studies for a multi–machine power system in Chapter 8.

Direct–current (DC) motors have found broad applications in industry because of their torque–speed characteristics. The significant features of DC drives include adjustable speed over a wide range, constant mechanical power output, rapid speed change and responsiveness to feedback signals. The speed of a DC motor can be varied by control of the field flux, the armature resistance, or the applied armature voltage. The three most commonly used speed control methods are shunt–rheostat control, armature circuit resistance control, and armature terminal voltage control[5–1].

In this chapter, only the most versatile speed control, that is, speed control by modulating the armature terminal voltage is discussed. In this method alone, speed control can be realized with different control strategies, one of which is PID control. PID stands for Proportional, Integral and Derivative. Such a controller is used to automatically adjust a certain variable so that the output of the controlled process will be held at some set–point. The difference between set–point and the actual measurement of that output is defined as error which is used to initiate the adjustment. This will be further covered in section 5.4 where performance studies are carried out.

5.2 Problem Formulation

Let us consider a separately excited DC motor[5–1] as shown in Figure 5.1. The generated speed voltage and the electromagnetic torque developed by the motor can be expressed as,

$$e_a = K\phi_d\omega_m = ki_f\omega_m \tag{5.1}$$

$$T_e = K\phi_d i_d = ki_f i_a \tag{5.2}$$

where,

K and k are constants,

ϕ_d is the direct axis air gap flux that is linearly proportional to the field current i_f,

ω_m is the angular velocity corresponding to the speed of rotation, and

i_a is the armature current.

The voltage equation for the field circuit is expressed as,

$$v_f = L_{ff}\frac{d}{dt}(i_f) + R_f i_f \tag{5.3}$$

where,

v_f, i_f, R_f and L_{ff} are the terminal voltage, current, resistance and self-inductance of the field circuit, respectively.

The voltage equation for the armature circuit is given by,

$$v_t = e_a + R_a i_a + L_{aq}\frac{d}{dt}(i_a) \tag{5.4}$$

where,

v_t, R_a, and L_{aq} are the terminal voltage, resistance and self-inductance of the armature circuit, respectively.

The dynamic equation for the mechanical system of the motor is expressed as,

$$T_e = ki_f i_a = J\frac{d}{dt}(\omega_m) + B\omega_m + T_L \tag{5.5}$$

or

$$T_e - T_L = J\frac{d}{dt}(\omega_m) + B\omega_m \tag{5.6}$$

where,

J is the combined polar moment of enertia of the load and the rotor of the motor,

B is the equivalent viscous friction constant of the load and the motor,

T_L is the mechanical load torque.

The Laplace transform of the voltage equation (5.4) for the armature circuit can be written as,

$$V_t - E_a = R_a(1+\tau_a s)I_a \tag{5.7}$$

where,

$\tau_a = L_{aq} / R_a$, the electrical time constant of the armature circuit.

The Laplace transform of (5.1) is given by,

$$E_a = L\{kI_f \cdot \omega_m\} \tag{5.8}$$

where L denotes Laplace transform. It is noticed that this is a nonlinear equation. When I_f is constant, it is easy to carry out the transform.

The Laplace transform of (5.6), with a zero initial condition, is given by,

$$T_e - T_L = Js\Omega_m + B\Omega_m \tag{5.9}$$

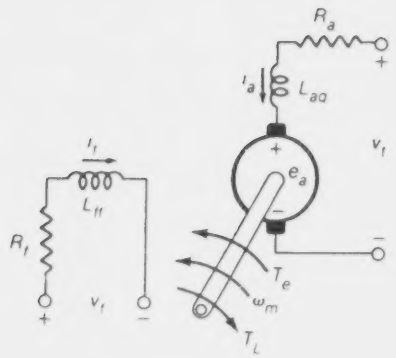


Fig. 5.1 Schematic Representation of A DC Motor[5–1]

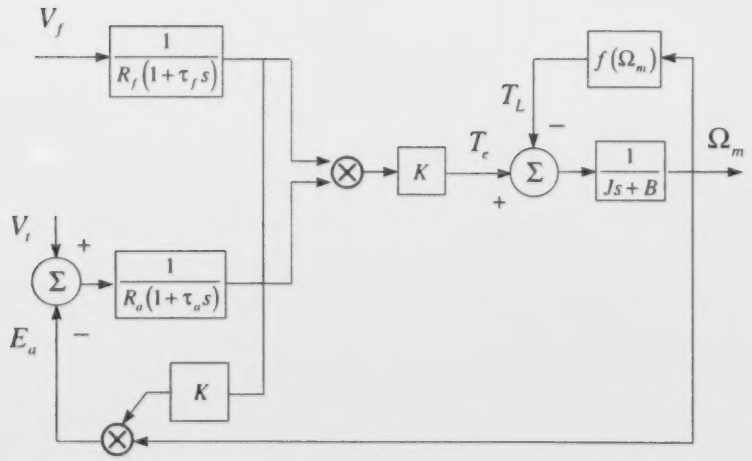


Fig. 5.2 Block Diagram of A DC Motor

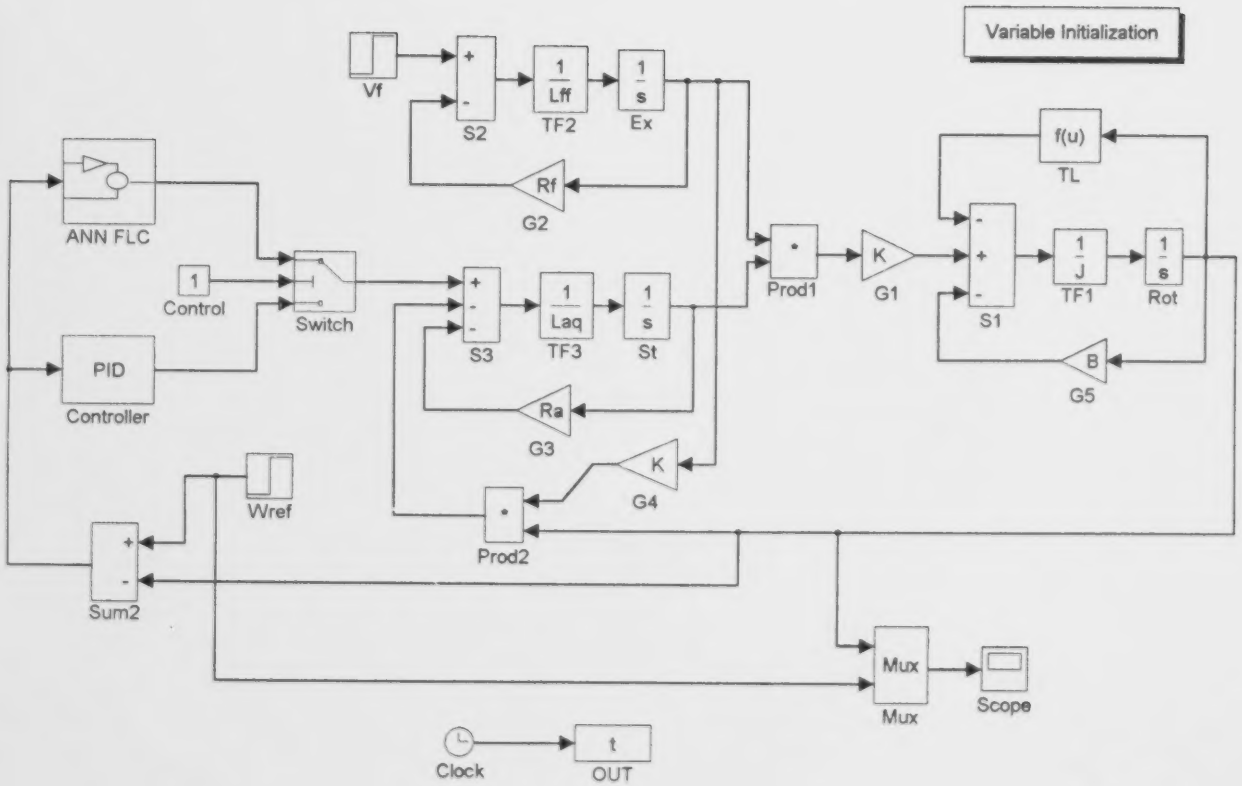


Fig. 5.3 Simulation Model of A DC Motor

Figure 5.2 shows the block diagram of the dynamics of a DC motor. Suppose that a 5–hp, 220V, separately excited DC motor has the following parameters: $R_a = 0.5\Omega, L_{aq} = 0.3H, k = 2H, R_f = 220\Omega, L_{ff} = 110H$. The combined constants of the motor armature and the load are $J = 3kg \cdot m^2$ and $B = 0.3kg \cdot m^2 / s$. The field current can be changed by changing the field voltage V_f. Using these data, a simulation model of the motor is constructed as shown in Figure 5.3. Let the step input voltage be 220V. Figure 5.4 shows the dynamic response. The steady state value of the motor speed is 107 rad/s.

5.3 Design of An ANN–FL Controller

5.3.1 Introduction

Based on the hybrid ANN–FL modeling technique presented in Chapter 4, an intelligent control system can be designed for speed control of the DC motor system. To accomplish such a design, the following information, to be discussed in this section, is needed.

(1) A knowledge base that consists of a rule base and a data base. The rule base defines a set of linguistic rules established on expert knowledge of the dynamic behavior of the fuzzy sets used in the rule base. Experience and engineering judgment is incorporated in designing the data base. This will be further covered in the following sections.

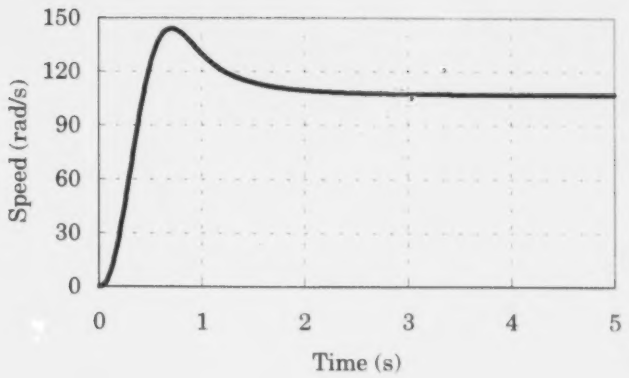


Fig. 5.4 Speed Response of the DC Motor with No Feedback Control

(2) Decision–making logic that imitates human's decision–making processes using fuzzy concepts.

(3) Fuzzification that transforms crisp variables into fuzzy sets. Defuzzification that does the opposite.

(4) An ANN–FL model as defined in Chapter 4.

(5) A training set of data that is collected either from real–world operation or computer simulation of the physical plant.

(6) A training program that implements the above algorithm.

5.3.2 Structure of the ANN–FL Controller

(1) *Linguistic Variables*

The goal of DC motor control in this study is to constrain the speed of the motor to a specified level. Suppose that the speed and its rate of change are

accessible and measurable. These two state variables can serve as inputs to the ANN–FL controller to be designed. The output of the controller is a change to the terminal voltage applied to the terminal of the amarture winding of the DC motor. Suppose that the state variables can be described by the following fuzzy sets, with v=very and m=moderate:

Change_of _speed =

[v_negative, m_negative, negative, positive, m_positive, v_positive]

Rate_of_speed_change =

[v_slow, m_slow, slow, fast, m_fast, v_fast]

The output variable of the controller can be described by a fuzzy set as follows:

Change_of_voltage = [negative, positive]

In fuzzy set theory, it would not matter how beautifully the linguistic variables are named, the control algorithm recognizes only what is happening in the physical process. Therefore, all the above fuzzy sets can be represented by linguistic labels such as negative big(NB), negative medium(NM), negative small(NS), positive small(PS), positive medium(PM), and positive big(PB). That is, these fuzzy variables can be expressed in terms of the corresponding linguistic labels as follows:

$$\Delta\omega = [NB, NM, NS, PS, PM, PB] \tag{5.10}$$

$$\Delta\omega / \Delta t = [NB, NM, NS, PS, PM, PB] \tag{5.11}$$

$$\Delta V = [N, P] \tag{5.12}$$

(2) *Membership Functions*

Suppose that the ranges of change of the fuzzy variables of the DC motor to be controlled are known from design or operational requirements. Then the universe of disclosure of the fuzzy sets can be determined. For example,

$$\Delta\omega = [-30,+30](rad) \tag{5.13}$$

$$\Delta\omega / \Delta t = [-75,+80](rad / s) \tag{5.14}$$

$$\Delta V = [-60,+60](volt) \tag{5.15}$$

Then, membership functions can be assigned to each of the above fuzzy sets, including both of those of the input and output variables. Figs. 5.5 – 5.7 show the membership functions for the state variable $\Delta\omega / \Delta t, \Delta\omega$, and the output variable ΔV, respectively. Membership functions for fuzzy sets $A_{i2},\ldots,A_{i5}(i = 1,2)$ are expressed by Gaussian function which takes the form of (5.16), and membership functions for fuzzy sets $A_{i1}, A_{i6}, B_i(i = 1,2)$ are expressed by a Bell function which takes the form of (5.17). Table 5.1 shows a rule base for the ANN–FL controller, where $A_{ij}, B_i(i = 1,2; j = 1,2,\ldots,6)$ are membership functions.

$$f(x,\sigma,c) = e^{-\left\{\frac{x-c}{\sqrt{2}\sigma}\right\}^2} \tag{5.16}$$

$$f(x,a,b,c) = \frac{1}{1+\left\{\frac{x-c}{a}\right\}^{2b}} \tag{5.17}$$

The parameters of these membership functions will be determined in a training process with data obtained from simulations of the DC motor with PID control.

(3) *Structure of the ANN–FL Controller*

A five–layer feedforward network, with two inputs and one output, is designed as shown in Figure 5.8. The universe of discourse of each input variable is initially divided into six segments. There are six membership functions for each of these variables. As has been mentioned earlier, each membership function is labeled with $A_{ij}(i = 1,2; j = 1,2,\ldots,6)$. The output variable has two membership functions, labeled as $B_i(i = 1,2)$. Definitions of these linguistic labels are given by (5.10) and (5.12).

The generalized Bell–curve function is chosen as the membership function of the output variable. Initially, A_{11} and A_{21} were represented by a Z–shaped membership function and A_{16} and A_{26} by a S–curve membership function. After training and simulation comparison, they are replaced with the generalized Bell–curve membership function. The rest of the membership functions are represented by Gaussian–curve function.

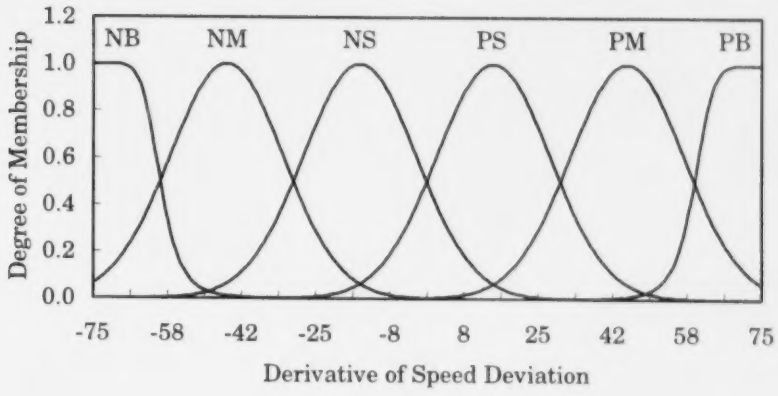


Fig. 5.5 Membership Functions for State Variable $\Delta\omega / \Delta t$

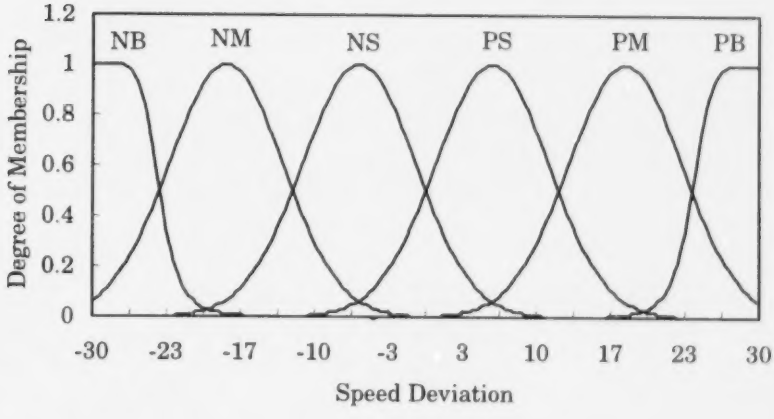


Fig. 5.6 Membership Functions for State Variable $\Delta\omega$

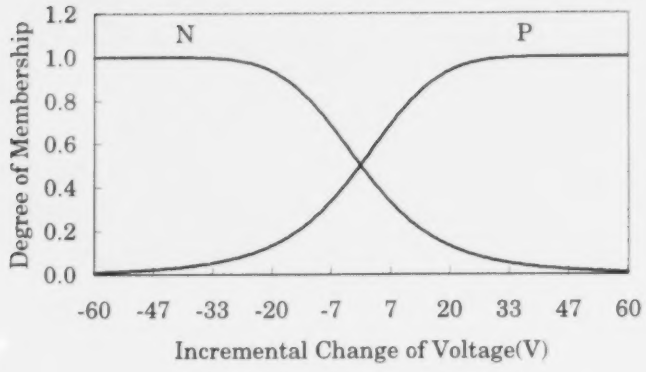


Fig. 5.7 Membership Functions for Output Variable ΔV

Table 5.1 Rule Base of the ANN–FL Controller

$\Delta\omega$ / ΔV / $\Delta\omega/\Delta t$	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
A_{21}	B_2	B_2	B_2			
A_{22}	B_2	B_2	B_2			
A_{23}	B_2	B_2	B_2			
A_{24}				B_1	B_1	B_1
A_{25}				B_1	B_1	B_1
A_{26}				B_1	B_1	B_1

(4) *Collection of Training Data*

A number of simulation studies of the DC motor dynamic system, shown in Figure 5.3, are carried out, simulating different load disturbances. These disturbances cause the changes of the state variables in their corresponding ranges as specified in (5.13) and (5.14). The available voltage change of the power source is specified in (5.15). The results from the simulations are recorded in Table 5.2. The first row in Table 5.2 is the speed deviation relative to reference speed. The first column is the derivative of the speed deviation. The numbers in Table 5.2 are the incremental voltage change relative to the normal terminal voltage which is 220 volts. These data will be used for training the ANN–FL controller. For design purpose, a near–optimal PID controller is used in the simulation to obtain the desired performance of the system. The parameters of the PID controller will be given later in this chapter when performance studies are carried out.

(5) *Supervised learning of the ANN–FLC system*

Up to this point, the ANN–FLC has an initial set of values for the parameters of the membership functions, given the fuzzy partitions and fuzzy logic rules. The task of a supervised learning scheme is to optimally adjust these parameters so that the controller will perform optimally no matter what the operating point is. Backpropagation [5–2] is used for the learning. A training program has been written using MATLAB and its Fuzzy Logic

Toolbox functionality. Another advantage of using this toolbox is that a newly–designed fuzzy controller can be easily interfaced with the TSSP package, see Chapter 7 of this thesis. The objective of the learning process is to find an optimal set of parameter values for the node functions in the network of Fig. 5.8 so that the total error function,

$$E_p = \frac{1}{2}\sum_{j=1}^{No}\left(t_{pj} - y_{pj}^{L}\right) \tag{5.18}$$

is minimized. Detailed learning rules have been presented in Section 4.4 of this thesis. There are 34 parameters for the membership functions, and 18 weights to be adjusted. The training set of data is given in Table 5.2. There are a total of 127 pairs of training patterns.

As the input variables are fuzzy to the controller, the stopping criterion for training doesn't have to be high. That means, training can be stopped at 1000 epochs or if the total error is less than, say, 0.2. After 1000 epochs of training, the network of the ANN–FL controller converges to a set of parameters that determine the membership functions and weights. These membership functions are displayed as per Figures 5.9–5.11. Figure 5.12 displays the RMSE(root mean squared error) curve of the training process. An adaptive procedure is used for changing the step size during training. The true step sizes used in the training are displayed in Figure 5.13. The reason for using a varying step size in each iteration is to ensure that the error function will not increase.

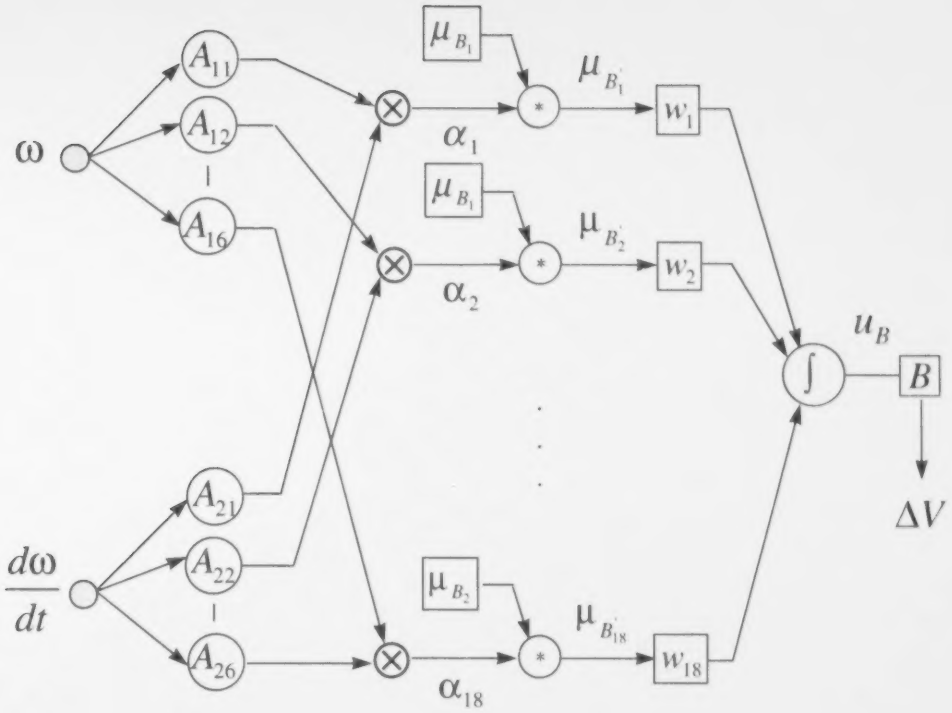


Fig. 5.8 An ANN-FL Controller for DC Motor Control

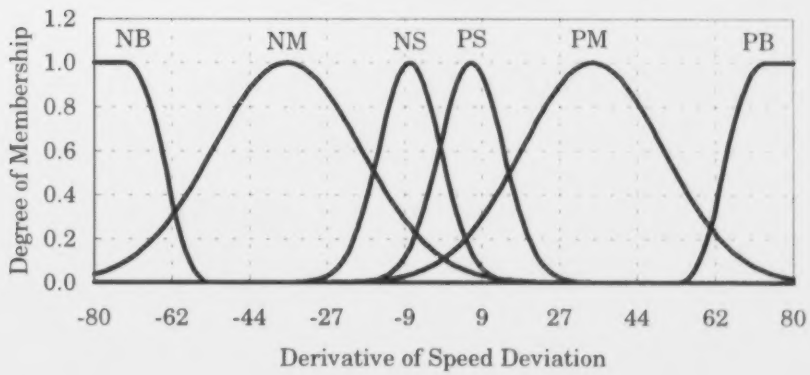


Fig. 5.9 Membership Functions for State Variable $\Delta\omega / \Delta t$

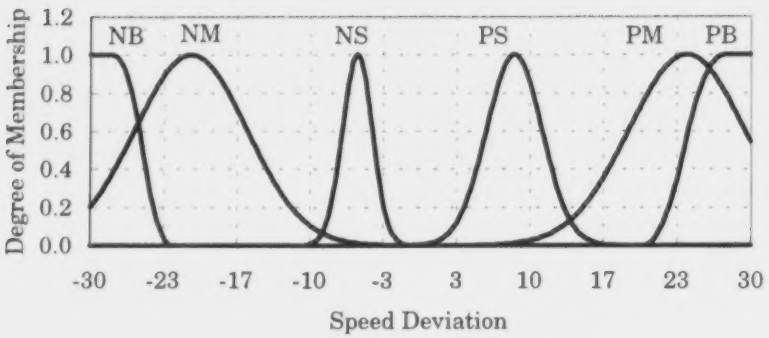


Fig. 5.10 Membership Functions for State Variable $\Delta\omega$

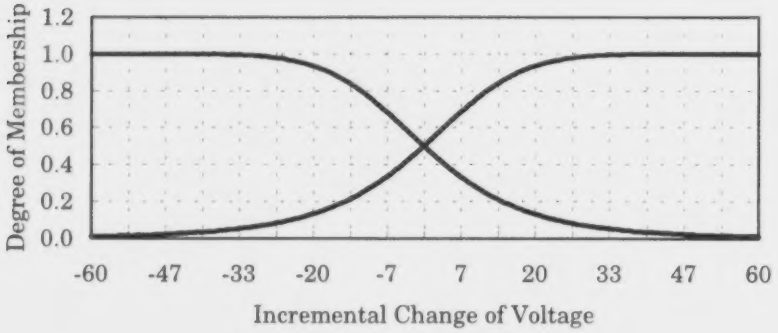


Fig. 5.11 Membership Functions for Output Variable ΔV

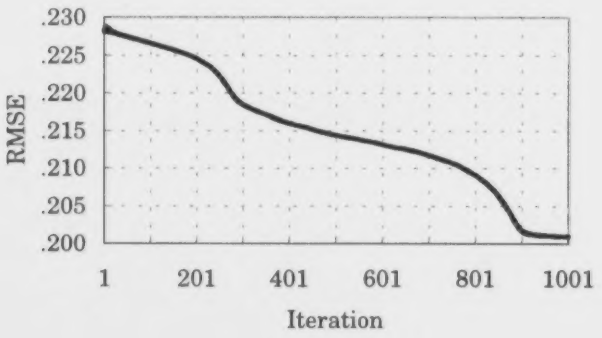


Fig. 5.12 RMSE Curve vs. Epochs

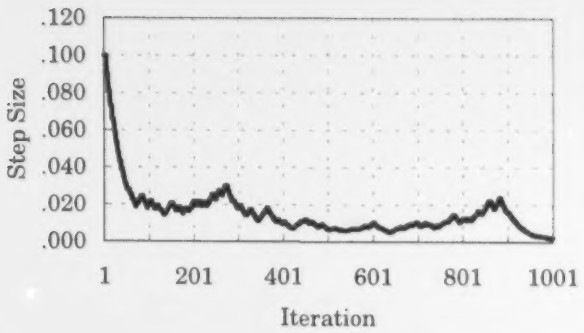


Fig. 5.13 Step Sizes vs. Epochs

Table 5.2 Training Data as Obtained from Simulation Studies

$\Delta\omega$ / ΔV / $\Delta\omega / \Delta t$	-28	-24	-19	-14	-9	4	-1	0	1	4	9	14	19	24	28
-75	60	55	50	46	42	38	34	30							
-62	55	50	46	41	37	33	29	25							
-49	50	46	40	36	32	28	24	20							
-36	46	41	36	30	27	23	19	15							
-23	42	37	32	27	20	16	13	10							
-10	38	33	28	23	16	10	7.5	5							
-1	34	29	24	19	13	7.5	2.5	1.25							
0	30	25	20	15	10	5	1.25	0	-1.25	-5	-10	-15	-20	-25	-30
5								-1.25	-2.5	-7.5	-13	-19	-24	-29	-34
13								-5	-7.5	-10	-16	-23	-28	-33	-38
28								-10	-13	-16	-20	-27	-32	-37	-42
41								-15	-19	-23	-27	-30	-36	-41	-46
54								-20	-24	-28	-32	-36	-40	-46	-50
67								-25	-29	-33	-37	-41	-46	-50	-55
80								-30	-34	-38	-42	-46	-50	-55	-60

5.4 PERFORMANCE EVALUATION

5.4.1 Introduction

As hardware and software technology has advanced a great deal in the last few years, digital control techniques have become more affordable and indispensable with the growing requirements for higher supervisory capability and better control performance in some control systems. For DC motors to achieve speed–tracking control, some adaptive control algorithms are required. Usually they are rather complicated and require a computer to do the job. Therefore, almost all applications of adaptive control to motor drives still remain in computer simulation studies for this reason[5–4]. The following study shows that a hybrid artificial neural network and fuzzy–logic(ANN–FL) controller needs only a simple and fast algorithm, once it is designed. This may open the door for developing fast and economical adaptive controllers for motor drives in the future.

In this section, a number of simulation studies will be presented of the DC motor dynamic system controlled by either a PID controller or the newly–designed ANN–FL controller. The gains of the PID controller are tuned to their optimal values at the normal operating point of the DC motor. In order to carry out a fair comparison, the two controllers will be kept unchanged during the entire simulation studies that follow. The investigation addresses three operating conditions, i.e., (1) operating at normal conditions with

different reference speeds; (2) operating at a speed of 110 radians per second or 525 revolution per minute, experiencing a load change or an excitation voltage change[5–4]; and (3) tracking a pre–specified trajectory[5–5].

5.4.2 Step Response with Different Reference Speeds

From earlier simulation studies, it is known that the DC motor under study operates at 107 radians per second at normal terminal voltage of 220 volts without any feedback control. When the DC motor is required to operate at another speed, it is necessary to have some sort of feedback control such as PID control. The PID controller used here has fixed design gains which are determined for a specific operating condition. Assume now that the DC motor operates at different set–points of speed, say from 70–130 radians per second. Extensive simulation studies were carried out and only a fraction of the obtained results are presented here. Figure 5.14(a) shows the situation where the reference speed, ω_{ref}, is 90 radians per second as applied in the simulation model of Fig. 5.3. By controlling the SWITCH, two step responses corresponding to the PID and the ANN–FL controller were obtained. By increasing the step input to 130 rad/s and following the same procedure, another two step responses were obtained as shown in Fig. 5.14(b). It can be concluded that the PID controller outperforms the ANN–FL controller for step responses.

(a) Step Input = 90 (rad/s)

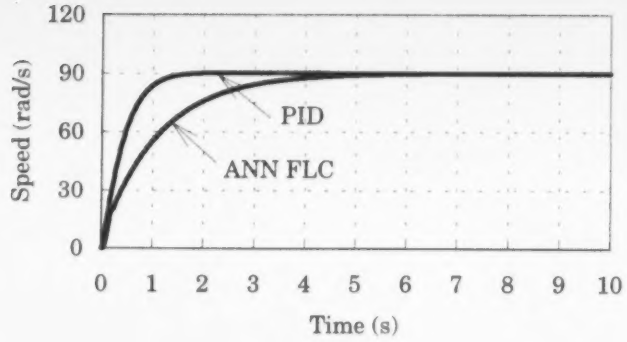


Fig. 5.14(a) Response of PID and ANN–FL Control to A Step Input of 90 rad/s

(b) Step Input = 130 (rad/s)

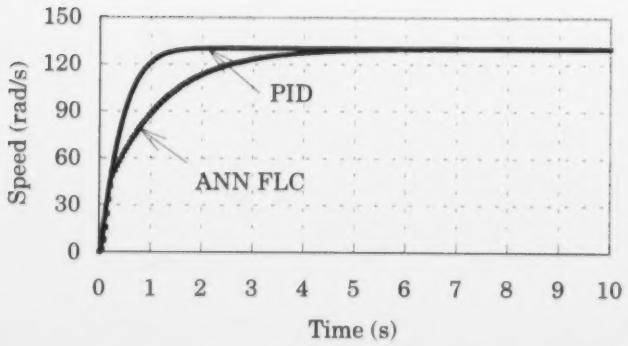


Fig. 5.14(b) Response of PID and ANN–FL Control to A Step Input of 130 rad/s

5.4.3 Different Disturbances

Two situations are considered, i.e., load torque disturbance and field voltage change. Load torque variation is present in any motor drive system, when a motor experiences a load torque increase, the rotor speed decreases. When the load torque decreases, the motor speed goes in the other direction. However, speed response to load torque disturbance is responsive when a feedback control is present. Field voltage variations result in a field current change which causes the emf change. In the end, it causes speed change.

Suppose that the DC motor is operating at a normal speed of 110 radians per second. Two scenarios were studied in this section for demonstrating the performance difference of the two controllers when the system withstands a certain disturbance. Scenario #1 involves load torque disturbance. Let a periodical load disturbance, shown in Fig. 5.15(a), be applied to the summation box S1 of the simulation model of the DC motor, shown in Fig. 5.3. Figure 5.15(b) shows the speed response of the motor to the disturbance, with the PID controller as feedback control. In this case the maximum speed ripple is 0.55 percent. Figure 5.15(c) shows the situation with the ANN–FL controller. The maximum speed ripple is now only 0.1 percent. Figure 5.16 shows a detailed comparison of the speed responses shown in Figs. 5.15(b) and 5.15(c). Note that the time–axis is stretched out so that transient details of the two controllers can be readily seen.

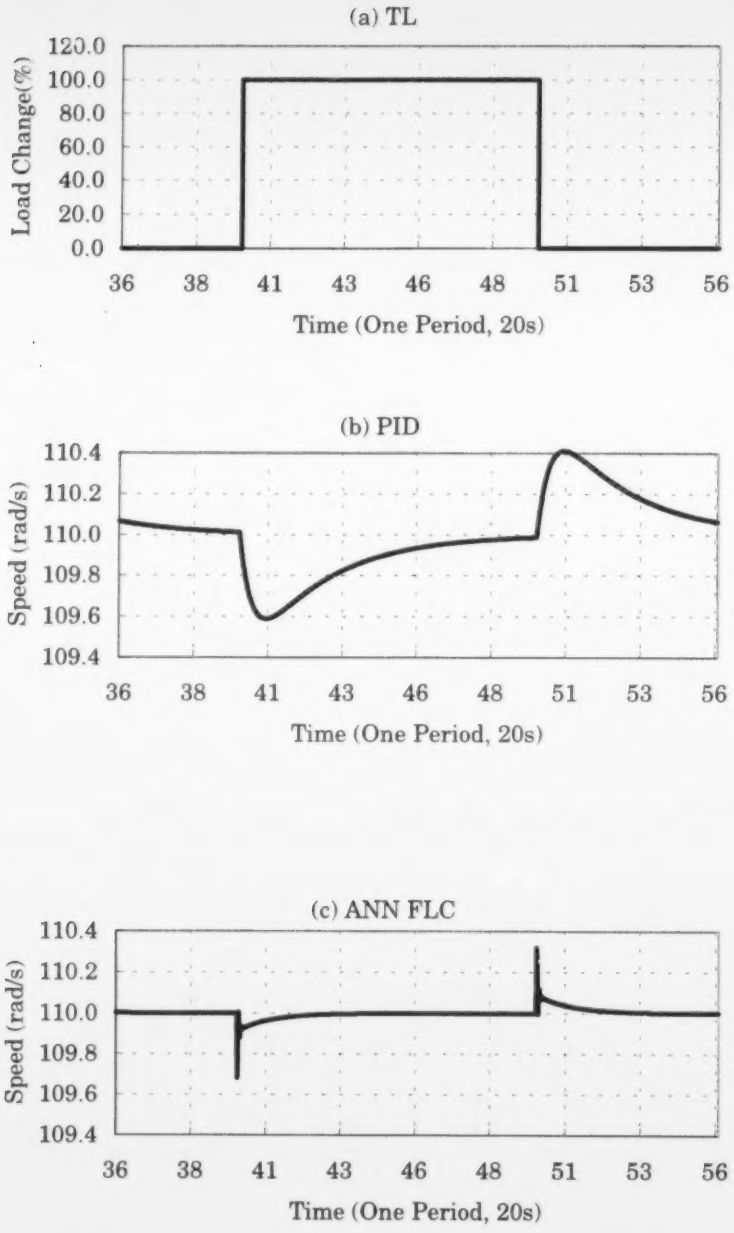


Fig. 5.15 Performance Comparison of ANN–FL vs. PID Control, with A Periodical Load Disturbance Applied

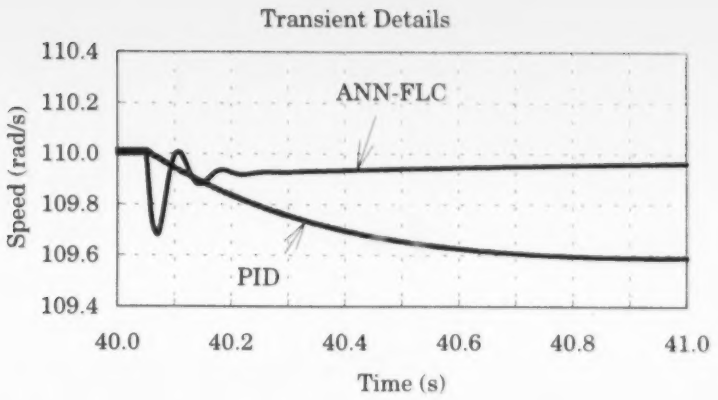


Fig. 5.16 Window Graphic Showing Transient Details in Figs. 5.15(b) & (c)

Scenario #2 involves field voltage change. A periodical voltage disturbance, shown in Fig. 5.17(a), is applied to the summation box S2 of Fig. 5.3. Figures 5.17(b) & (c) show the speed responses of the system with the ANN–FL controller as feedback control. The maximum speed fluctuations are 6, and 1 percent, respectively. Figure 5.18 shows the transient details of the two responses in this case.

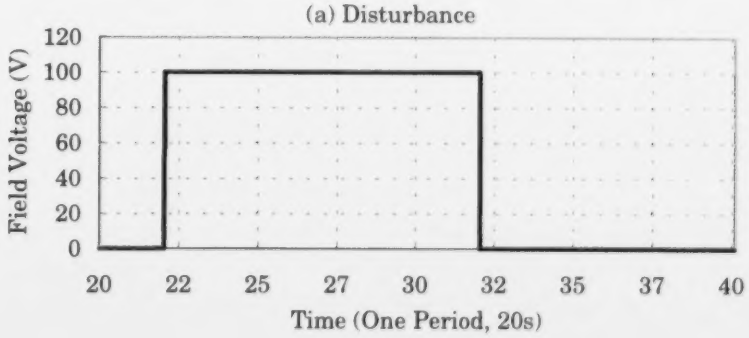
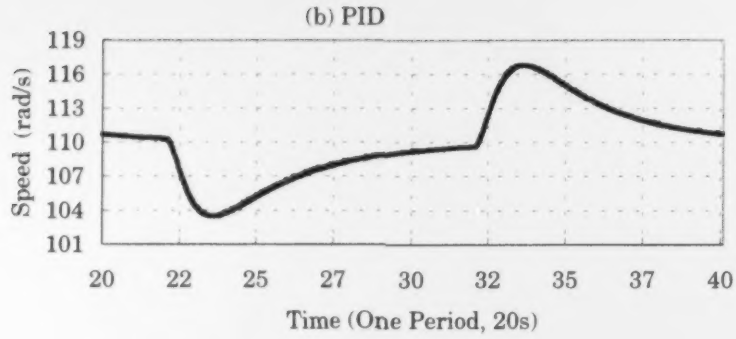
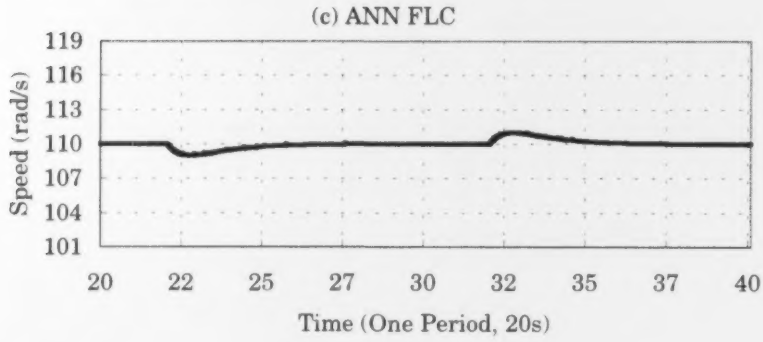


Fig. 5.17(a) A Periodical Field Voltage Disturbance





Figs. 5.17(b)(c) Performance Comparison of ANN–FL vs. PID Control Responding to the Disturbance of Fig. 5.17(a)

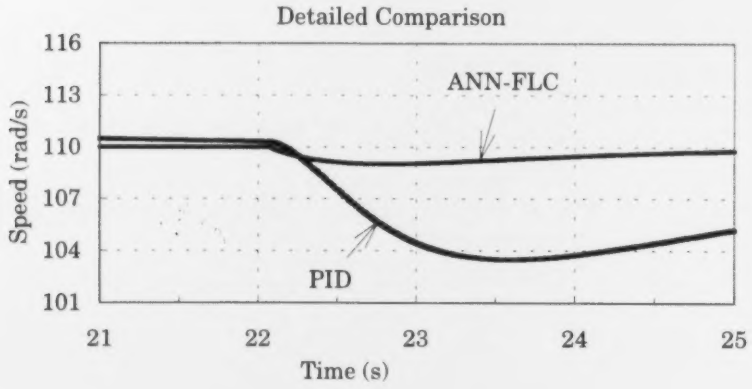


Fig. 5.18 Window Graphic Showing Transient Details in Figs. 5.17(b) & (c)

5.4.4 Trajectory Control

The objective of this simulation study is to demonstrate which of the two controllers can drive the DC motor to follow a pre–specified trajectory more closely. Suppose that the pre–specified speed trajectory is defined by

$$\omega_{ref} = 10\sin(2\pi f_1 t) + 16\sin(2\pi f_2 t) \tag{5.19}$$

where $f_1 = 1/4\,Hz$, $f_2 = 1/7\,Hz$ and $t = 30\,s$.

The simulation was carried out as follows. First, the pre–specified trajectory was computed and sampled with a sampling time step of 0.01 second over a time interval of [0, 30]. Then, it was used as the reference speed, ω_{ref}, in the simulation, using the model of Fig. 5.3. Two simulations were performed using either the PID or the ANN–FL controller. Figure 5.19(a) illustrates the situation where the PID is used. It can be seen that there is a maximum of 15 percent discrepancy between the desired speed and the actual response. Note that the manipulated variable is the output of the controller which is the input to the DC system. This variable is restricted in the range $(-60,+60)$ volts.

Figure 5.19(b) illustrates the case where the ANN–FL controller is used. It produces perfect speed–tracking control. The manipulated variable is also shown in the graphic. Several other different types of reference trajectories were simulated(results not shown). In each of these simulations, the newly–designed ANN–FL controller demonstrated much superior performance.

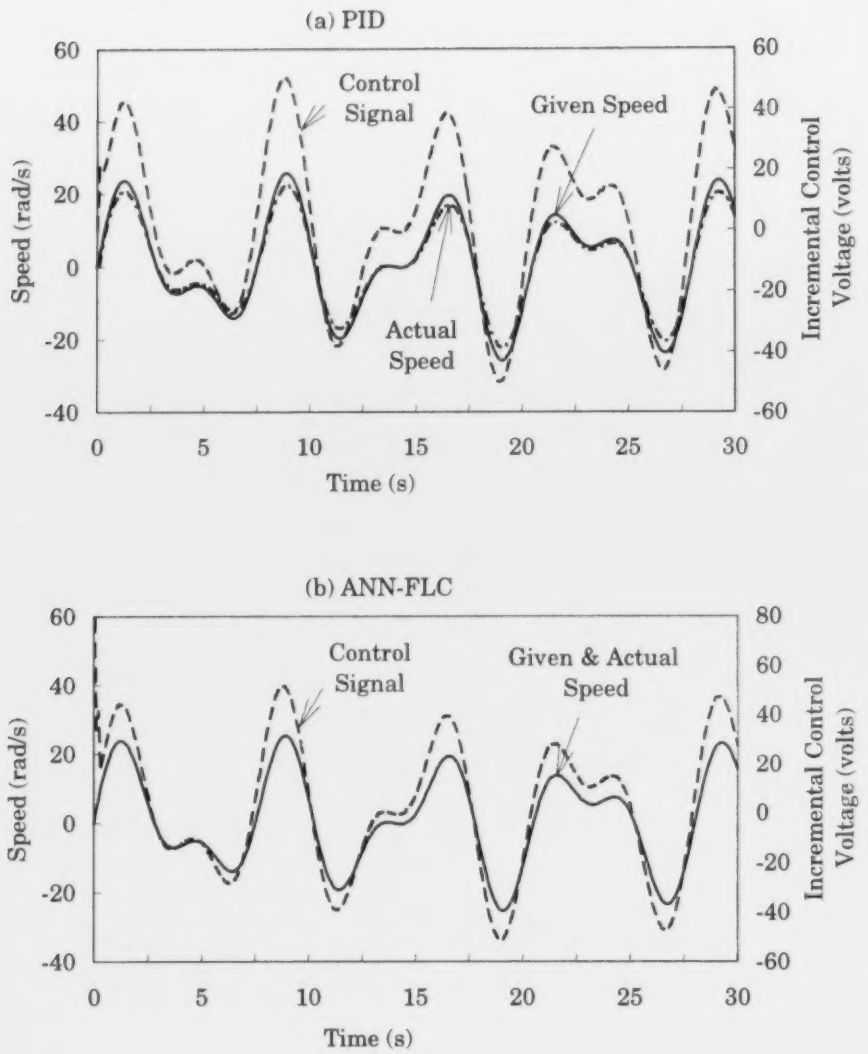


Fig. 5.19 Trajectory Tracking Control of the DC Motor by

(a)PID controller or (b)ANN–FL Controller

CHAPTER 6

MODELING OF COMPOSITE POWER SYSTEMS

The stability problem in a power system is principally one of keeping the interconnected synchronous generators in synchronism and damping rotor oscillations following a system disturbance. Therefore, accurate modeling of synchronous generators is of fundamental importance for off–line simulation studies related to the problem. A synchronous generator includes three parts; namely, a generating unit, an excitation system, and a speed–governing and turbine system. Synchronous generators are interconnected through an AC/DC transmission network to which various loads, such as induction motors, are connected as the end users of electrical energy. This chapter discusses the modeling aspect of these devices.

6.1 INTRODUCTION

The study of transient stability of a power system is concerned with the investigation of changing phenomena of electric quantities when the operating environment of the system is disturbed. To carry out such an investigation, power apparatuses must be modeled using mathematical expressions in computer simulations. They include synchronous machines, transformers, transmission lines, and loads. Synchronous machines play a

vital role in a power system. Synchronous generators convert mechanical energy into electric energy. Synchronous motors drive loads and convert electric energy to mechanical energy. Synchronous condensers generate reactive power into the system to improve voltage profiles. Transformers and transmission lines are equipment indispensable in transporting electric energy from the source to the point of consumption. Electric loads are variables that change over time. The principal task of the power system is to continuously match the electric energy supply to these changing loads, which gives rise to the problem of stability.

Large power systems consist of many local power systems connected through long transmission lines. In special situations, such connections may not be possible by using AC transmission. For example, two AC systems with different operating frequencies can not be connected with an AC link. In such situations high–voltage direct–current (HVDC) transmission has to be used.

6.2 Modeling of Power Components

6.2.1 Synchronous Machines

The actual structure of a synchronous machine is complicated. Magnetically it is equivalent to many windings coupled together. The synchronous machine under study is assumed to have three identical, symmetrically placed stator windings, one field winding and two amortisseur or damper

windings. All these windings are magnetically coupled, depending on the position of the rotor. Therefore the flux linking each winding is a function of the rotor position.

The mathematical description of the machine is very complicated as it consists of both stationary and rotational parts. Park transformation is usually used to simplified this description[6–1].

This transformation projects the original variables onto a rotational coordinate system called the $d-q-0$ system. When this transformation is performed, the flux equations of the numerous axes become linear functions of currents flowing in the stator and damper windings. A pictorial representation of the synchronous machine is given in Figure 6.1. Assume that positive stator currents i_d and i_q are generated currents, positive rotor currents i_f,i_D, and i_Q flow into the machine, where d,q,f,D and Q denote the direct–axis stator winding, quadrature–axis stator winding, and the field winding, the direct–axis and quadrature–axis damper windings. The voltage equations of the various windings are listed below, with transformer voltages ignored,

$$v_d = -\frac{d}{dt}(\phi_q) - r_a i_d \tag{6.1}$$

$$v_q = \frac{d}{dt}(\phi_d) - r_a i_q \tag{6.2}$$

$$\frac{d}{dt}(\phi_f) = v_f - r_f i_f \tag{6.3}$$

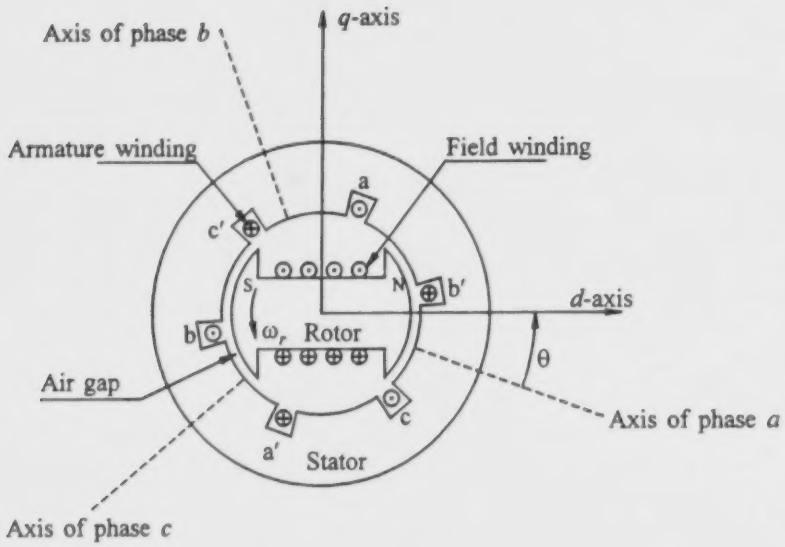


Fig. 6.1 Pictorial Representation of A Synchronous Machine

$$\frac{d}{dt}(\phi_D) = -r_D i_D \tag{6.4}$$

$$\frac{d}{dt}(\phi_Q) = -r_Q i_Q \tag{6.5}$$

where,

v = the voltage,

r = the resistance, and

ϕ = the flux linkage of a winding.

The flux linkage is determined by the following equation, accordingly,

$$\phi_d = -x_d i_d + x_{ad} i_f + x_{ad} i_D \tag{6.6}$$

$$\phi_f = -x_{ad} i_d + x_f i_f + x_{fD} i_D \tag{6.7}$$

$$\phi_D = -x_{ad} i_d + x_{fD} i_f + x_D i_D \tag{6.8}$$

$$\phi_q = -x_q i_q + x_{aq} i_Q \tag{6.9}$$

$$\phi_Q = -x_{aq} i_q + x_Q i_Q \tag{6.10}$$

where,

x_d, x_q, x_{fD}, x_D and x_Q are complete reactance,

x_{ad} and x_{aq} are mutual reactance, and

$x_{fD} = x_{ad}$.

Solving (6.6) ~ (6.8) for ϕ_d expressed in terms of ϕ_f, ϕ_D and i_d, we have

$$\begin{aligned}\phi_d = {} & \frac{x_{ad}}{x_D x_f - x_{ad}^2}\left\{\left(x_D - x_{ad}\right)\phi_f + \left(x_f - x_{ad}\right)\phi_D\right\} \\ & - \left\{x_d - \frac{x_{ad}^2\left(x_D + x_f - 2x_{ad}\right)}{x_D x_f - x_{ad}^2}\right\} i_d\end{aligned} \tag{6.11}$$

Defining the $q-$axis subtransient voltage as,

$$E_q'' = \frac{x_{ad}}{x_D x_f - x_{ad}^2}\left\{\left(x_D - x_{ad}\right)\phi_f + \left(x_f - x_{ad}\right)\phi_D\right\} \tag{6.12}$$

and subtransient reactance as,

$$x_d'' = \left\{x_d - \frac{x_{ad}^2\left(x_D + x_f - 2x_{ad}\right)}{x_D x_f - x_{ad}^2}\right\} \tag{6.13}$$

Then the voltage equation of (6.2) can be rewritten as,

$$v_q = E_q'' - x_d'' i_d - r_a i_q \tag{6.14}$$

Similarly, defining the d – axis subtransient voltage and reactance as,

$$E_d'' = -\frac{x_{aq}}{x_Q}\phi_Q \tag{6.15}$$

$$x_q'' = x_q - \frac{x_{aq}^2}{x_Q} \tag{6.16}$$

Then the voltage equation of (6.1) on d – axis can be rewritten as

$$v_d = E_d'' + x_q'' i_q - r_a i_d \tag{6.17}$$

A block diagram for this model is shown in Figure 6.2.

(a) *Two–Axes Model with Subtransient*

In this model, the damping effect of the rotor is represented by two short–circuited damper windings. The following derived differential equations are utilized to describe the subtransient behavior of a synchronous machine (see Appendix A for detailed derivation):

$$\frac{d}{dt}\left(E_q'\right) = \frac{1}{\tau_{do}'}\left\{kE_{fd} - E_q' - \left(x_d - x_d'\right)i_d\right\} \tag{6.18}$$

$$\frac{d}{dt}\left(E_{sum}\right) = \frac{1}{\tau_{do}''}\left\{-E_{sum} - \left(x_d' - x_d''\right)i_d\right\} \tag{6.19}$$

$$\frac{d}{dt}\left(E_d''\right) = \frac{1}{\tau_{qo}''}\left\{-E_d'' + \left(x_q - x_q''\right)i_q\right\} \tag{6.20}$$

where $E_{sum} = E_q'' - E_q'$.

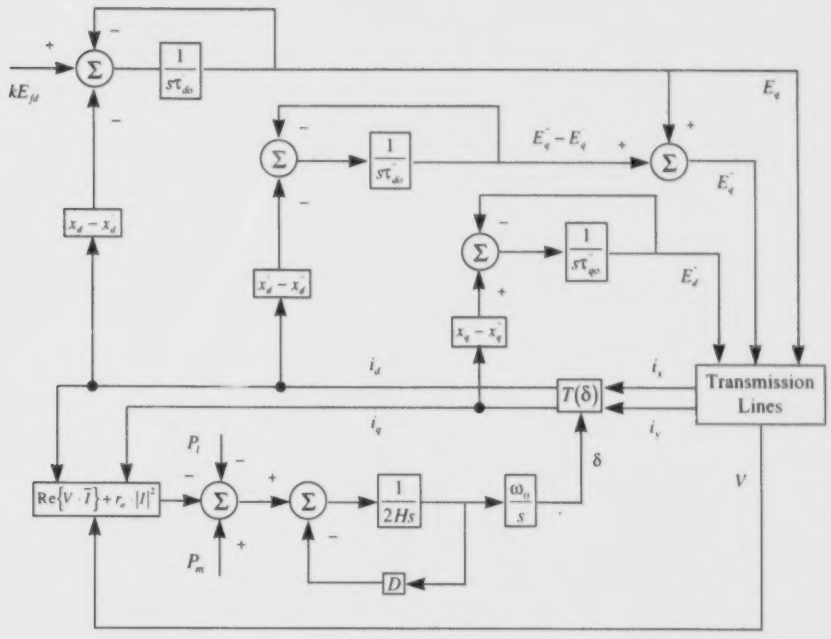


Fig. 6.2 Block Diagram of the Synchronous Generator Model

The differential equation describing machine motion is given by

$$\frac{d^2\delta}{dt^2} = (P_m - P_e - P_l - D\Delta\omega)/2H \tag{6.21}$$

$$\frac{d\delta}{dt} = \omega_0 (\Delta\omega) \tag{6.22}$$

where

H = the machine inertia constant in seconds,

P_m = the mechanical power input to the generator,

P_e = the electrical power output from the generator to the system,

P_l = the power loss in the stator windings, and

ω_0 = the synchronous speed of the machine in radians per second.

Generator saturation can be taken into account by modifying the reactance x_{ad} and x_{aq} according to machine saturation at 1.0 per unit and 1.2 per unit terminal voltage that usually are known from the manufacturer. There are several algorithms to realize this modification [6–2,6–3]. An iterative procedure was employed in the studies reported in this thesis. At the same time, the corresponding time constants must also be modified. A detailed derivation of these expressions is given in Appendix. The equations are included here to maintain completeness of the mathematical model of the synchronous machine modeled by the two–axis representation with subtransient. The saturated time constants are modified as follows:

$$\tau'_{do} = \tau'^{(0)}_{do}\left\{k + (1-k)\frac{x'_d - x_l}{x'^{(0)}_d - x_l}\right\} \tag{6.23}$$

$$\tau''_{do} = \tau''^{(0)}_{do}\left\{k + (1-k)\frac{x''_d - x_l}{x'^{(0)}_d - x_l}\right\} \tag{6.24}$$

$$\tau''_{qo} = \tau''^{(0)}_{qo}\left\{k + (1-k)\frac{x''_q - x_l}{x'^{(0)}_q - x_l}\right\} \tag{6.25}$$

where the superscript (0) indicates unsaturated values, and k is the saturation factor. Other symbols are listed at the beginning of this thesis.

The interface between the generator model and the terminal of the transmission network is accomplished by the following two equations.

$$v_q = E''_q - x''_d i_d - r_a i_q \tag{6.26}$$

$$v_d = E''_d + x''_q i_q - r_a i_d \tag{6.27}$$

(b) *Two–Axis Model with Transient*

This model is similar to the one presented previously. The subtransient effects are totally ignored, however, the transient effects are taken into account. There are two rotor windings, one is the field winding in the $d-$axis and the other is the equivalent damper winding in the $q-$axis formed by the solid rotor. The basic equations of this model are given by:

$$v_q = E_q^{'} - x_d^{'} i_d - r_a i_q \tag{6.28}$$

$$v_d = E_d^{'} + x_q^{'} i_q - r_a i_d \tag{6.29}$$

$$\frac{d}{dt}\left(E_q^{'}\right) = \frac{1}{\tau_{do}^{'}}\left\{kE_{fd} - E_q^{'} - \left(x_d - x_d^{'}\right)i_d\right\} \tag{6.30}$$

$$\frac{d}{dt}\left(E_d^{'}\right) = \frac{1}{\tau_{qo}^{'}}\left\{-E_d^{'} + \left(x_q - x_q^{'}\right)i_q\right\} \tag{6.31}$$

where $\tau_{do}^{'}$ is modified according to (6.24); and

$$\tau_{qo}^{'} = \tau_{qo}^{'(0)}\left\{k + (1-k)\frac{x_q^{'} - x_l}{x_q^{(0)} - x_l}\right\}. \tag{6.33}$$

(c) *One–Axis Model with Transient*

In this model, the amortisseur effects are totally neglected. There is only one rotor winding present in the model, i.e., the field winding. Its differential equation is given by (6.34). Saturation is treated in the same manner as explained previously. The difference between this model and the two–axis transient model is that as no damper windings are modeled, the differential equation for $E_d^{'}$ is, therefore, eliminated.

$$\frac{d}{dt}\left(E_q'\right) = \frac{1}{\tau_{do}'}\left\{kE_{fd} - E_q' - \left(x_d - x_d'\right)i_d\right\} \tag{6.34}$$

(d) *Infinite–Bus Model*

To accommodate a large power system in computer simulation studies, it is often necessary that part of the system be modeled as an infinite–bus which generates or absorbs large amount of power and at the same time maintains constant terminal voltage. Details on how this is done will be given in Chapter 7.

6.2.2 Excitation Systems

Since the 1960s, excitation systems have been represented in extra detail in transient stability programs, thanks to the availability of economic computing power. A 1968 IEEE report reviewed the computer representation of excitation systems available at the time[6–4]. In 1981, IEEE published standard excitation models used in the power industry[6–5]. In 1992, IEEE Standard 421.5, "IEEE Recommended Practice for Excitation System Models for Power System Stability Studies" was adopted[6–6]. These models are also used in TSSP software[6–7]. In recent years, there have been many new excitation systems applied. Most of the new systems have digital–based controls, that is, they use microprocessor technology to implement the control algorithms. These digital–based controls offer flexibility and control options that are difficult to implement in analog control systems.

Standard analog excitation systems are listed in [6–5, 6–6] and user–defined models can be incorporated in the TSSP software[6–7]. Some variations of these standard models are presented in[6–8]. In this section, we present two analog and one digital excitation system. Figure 6.3 shows the block diagram of an analog excitation system, referred as DC Type 1 excitation system[6–5], where $V_{err} = V_{ref} - V_t$; V_s is the output signal from a power system stabiliser.

Figure 6.4 shows the general structure of a digital excitation system. It consists of the digital part and the interface A/D and D/A converters as included in the dashed–line box. The output of the generator is sensed and scaled down to appropriate levels and converted into digital words by the A/D converter. It is then compared with the reference input that is held in the RAM of the microprocessor. The result is acted upon by a controller that can realize classical control algorithms such as PID control and newer schemes such as fuzzy logic and adaptive control. Apart from the common features available in both analog and digital excitation systems, the latter can be designed to implement extremely sophisticated control algorithms such as transient gain reduction, power system stabilizer(PSS) control, Var/power factor control, under/over excitation limit control, stator current limit control and many other features[6–8]. Digital systems are relatively immune to parameter changes. They provide increased accuracy and precision.

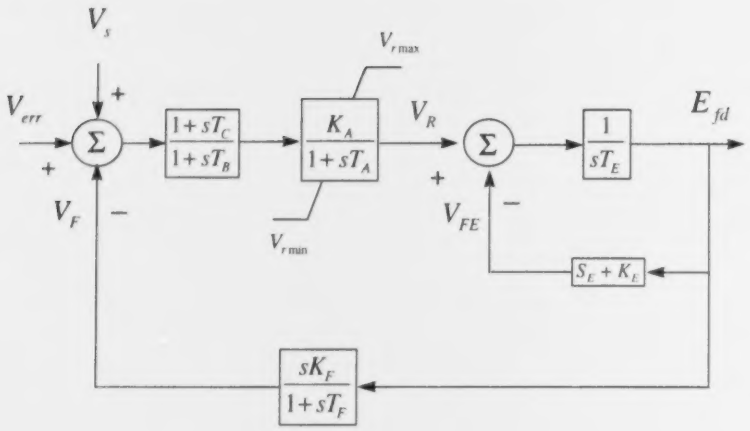


Fig. 6.3 Block Diagram of DC Type 1 Excitation System[6–5]

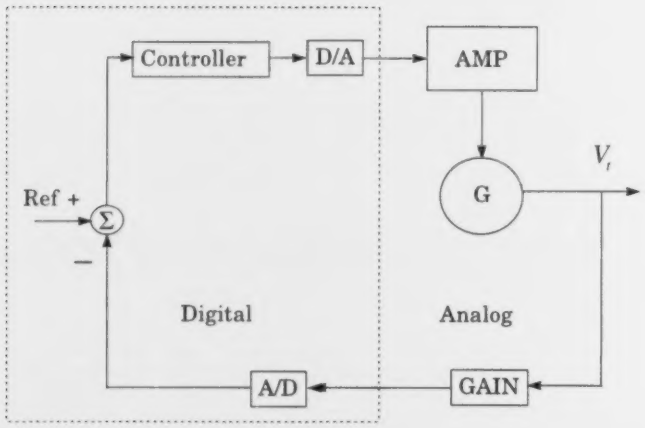


Fig. 6.4 Block Diagram of A Digital Excitation System

Shown in Figure 6.5 is an alternator–rectifier excitation system. It represents the Basler DECS voltage regulator as applied to a brushless exciter[6–6]. The AVR in this model consists of PID control. The values for the proportional, integral, and derivative gains are chosen for best performance from each particular generator excitation system. Other excitation models will be further covered in Chapter 7 where TSSP in SIMULINK is presented.

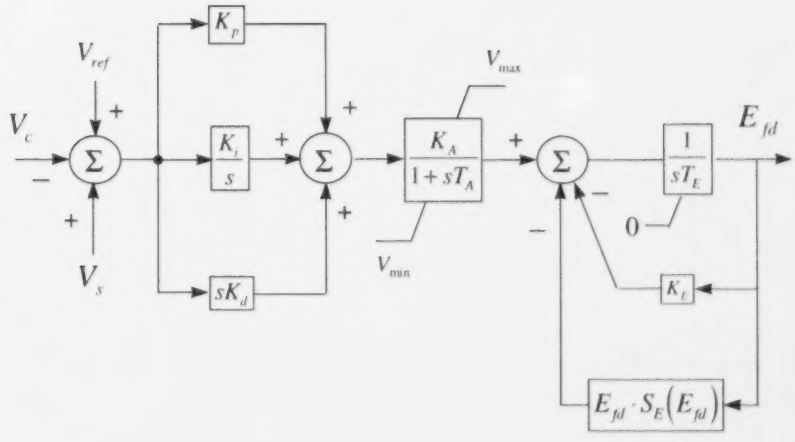


Fig. 6.5 Alternator–Rectifier Excitation System[6–6]

6.2.3 PRIME MOVERS

Electrical energy is converted from mechanical energy by synchronous generators. The prime mover governing system provides a means of controlling the input of mechanical energy to the generator so that the generated power and its frequency can be controlled. This prime mover includes speed–governing and turbine system for both steam and hydro generators. In this section, we discuss both hydro and steam turbines and governing systems. For detailed description of these systems, the reader is referred to [6–10].

There are two types of speed governing systems for hydroturbines, i.e., mechanical–hydraulic control and electric–hydraulic control. As the dynamic performance of the electric governor is necessarily adjusted to be the same as that of the mechanical governor for interconnected system operation, only one single model is needed[6–11]. Figure 6.6 shows the detailed model of a hydro–turbine and speed governing system. Figure 6.7 shows its equivalent representation. Their simulation models will be discussed in Chapter 7.

Figure 6.8 shows a block diagram that may be used to represent either a mechanical–hydraulic system or an electric–hydraulic system for a steam turbine by selecting the parameters appropriately. In the model it is seen that P_0 is an initial load reference. Figure 6.9 shows the Tandem–Compound Double Reheat steam turbine system[6–11].

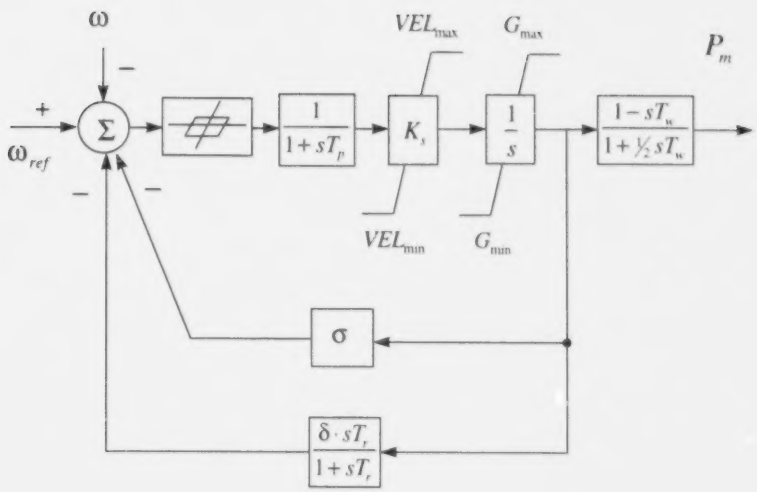


Fig. 6.6 Detailed Representation of Hydrogovernor Turbine System[6–11]

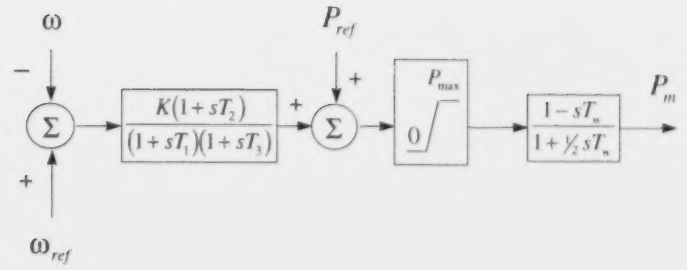


Fig. 6.7 Equivalent Representation of Hydrogovernor Turbine System[6–11]

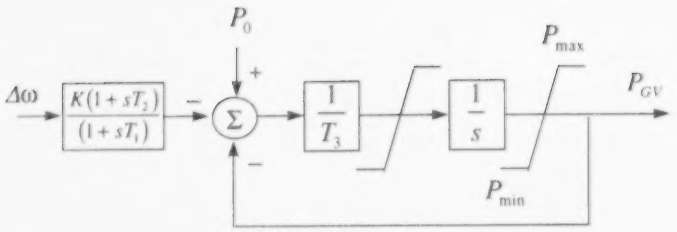


Fig. 6.8 Speed Governing System for Steam Turbines[6–11]

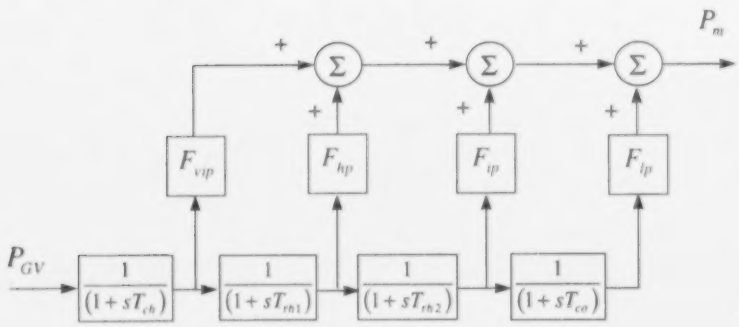


Fig. 6.9 Block Diagram of A Steam Turbine[6–11]

6.2.4 Conventional PSS

In order to suppress the low frequency electromechanical oscillations in a power system, supplementary excitation controllers known as power system stabilizers(PSSs) are often installed to provide positive damping [6–12]. The block diagram of the IEEE standard PSS model[6–5] is shown in Figure 6.10. The transfer function of the filter in the diagram can be expressed by

$$\frac{1+A_5 s+A_6 s^2}{\left(1+A_1 s+A_2 s^2\right)\left(1+A_3 s+A_4 s^2\right)} \tag{6.35}$$

The input signal to a PSS can be arbitrary. Generally, any one or a combination of the following can be utilized as input signal. (a) deviation of machine shaft speed; (b) deviation of terminal frequency; (c) net accelerating power; (d) deviation of terminal voltage.

The input signals must be in per unit. A PSS with a different input signal will have a different transfer function. Two specific PSS models are used in the studies reported in this thesis. Their input signals are generator electrical power and net accelerating power, respectively. The former is identified as IEEEST and the latter as IEEESN. Their block diagrams are shown in Figure 6.11 and Figure 6.12, respectively.

Note that only a single time constant A_1 is present in the filter transfer function in Figs. 6.11 and 6.12. This is because of the inherently low level of torsional interaction when net accelerating power is used as stabilizer input.

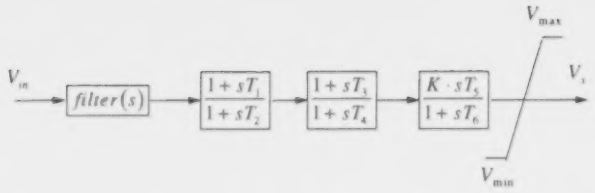


Fig. 6.10 IEEE Standard PSS Model[6–12]

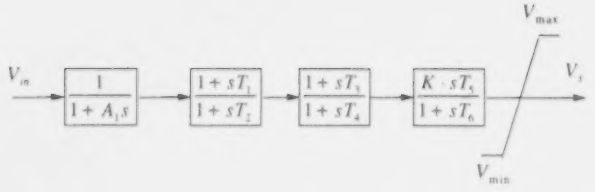


Fig. 6.11 IEEEST PSS Using Net Accelerating Power As Input[6–5]

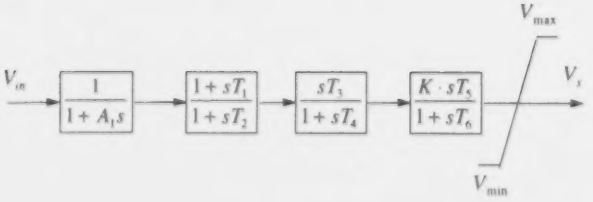


Fig. 6.12 IEEE PSS Using Generated Electric Power As Input[6–5]

6.2.5 Power System Loads

Appropriate representation of power system loads in transient stability studies is important as loads have a great influence on system stability. On the other hand, the modeling of loads is a complicated issue as there are so many loads in a system and at each bus the load composition is different and most of the time is unknown. Moreover, this composition changes over time. Therefore, simplification and certain assumptions have to be used in representing each load. In [6–13], IEEE has presented the state of the art of representation of power system loads for dynamic performance analysis in practice today. In this section, we present the most frequently used load models in transient stability studies. Detailed modeling of large induction motors is also given [6–14].

(1) *Constant Impedance Model*

In this model, the power varies directly with the square of the voltage magnitude, i.e.,

$$P + jQ = (1/Z) \cdot V^2 \tag{6.36}$$

(2) *Constant Current Model*

In this model, the power varies directly with the voltage magnitude, i.e.,

$$P + jQ = V \cdot \bar{I} \tag{6.37}$$

(3) *Constant Power Model*

In this model, the power does not vary with the changes in voltage magnitude. It may also be called constant MVA load model.

(4) *Polynomial Load Model*

This model expresses the relationship between power and voltage magnitude as:

$$P = P_0\left\{a_1\left(V/V_0\right)^2 + a_2\left(V/V_0\right) + a_3\right\} \tag{6.38}$$

$$Q = Q_0\left\{a_4\left(V/V_0\right)^2 + a_5\left(V/V_0\right) + a_6\right\} \tag{6.39}$$

where

V_0 = the rated voltage,

P_0 = the active power consumed at rated voltage;

Q_0 = the reactive power consumed at rated voltage.

However, these values are normally taken as the values at the initial system operating condition for the study. The coefficients in the model satisfy the following equations[6–13]:

$$a_1 + a_2 + a_3 = 1 \tag{6.40}$$

$$a_4 + a_5 + a_6 = 1 \tag{6.41}$$

6.2.6 Induction Motors

Representation of induction motors for transient stability studies can be derived following similar procedures as used in deriving synchronous machine models. The final model can be expressed in the $d-q-0$ reference frame. As the rotor of an induction motor has a symmetrical structure, its equivalent circuit on $d-$axis, and $q-$axis will be the same. The dynamics of the rotor circuits are determined by the slip that changes with the loading. In this section, we present detailed modeling of induction motors. The material that follows logically belongs to the section of power system loads. The present arrangement emphasizes its importance and uniqueness as compared with other loads.

For a three–phase induction motor, the number of poles of the field produced by the induced rotor currents is the same as that of the field produced by the stator winding. Therefore the rotor can be modeled by an equivalent three–phase winding. The rotor may be constructed in one of two ways, i.e., one with wound rotor where conventional three–phase windings are brought out through three slip rings on the shaft and connected to external circuits; the other with squirrel–cage rotor which consists of a number of bars short–circuited by end rings at both ends.

Choose the $d-q-0$ reference frame with its axes rotating at the synchronous speed. The $q-$axis is 90° ahead of the $d-$axis in the direction of rotation. Let the $d-$axis coincide with the phase $a-$axis at time $t \geq 0$. As

indicated in Figure 6.13, its displacement from phase $a-$axis at any time t is $\omega_s t$, where ω_s is the angular velocity of the stator field in electrical radians per second.

Let subscript s denote quantities of the stator and r of the rotor. The mathematical equations of an induction motor in per unit in the $d-q-0$ reference frame, can be summarized as follows[6–10]:

Stator voltage:

$$v_{ds} = R_s i_{ds} - \omega_s \phi_{qs} \tag{6.42}$$

$$v_{qs} = R_s i_{qs} + \omega_s \phi_{ds} \tag{6.43}$$

Rotor voltages:

$$v_{dr} = R_r i_{dr} - \frac{d}{dt}(\theta_r)\phi_{qs} + \frac{d}{dt}(\phi_{dr}) \tag{6.44}$$

$$v_{qr} = R_r i_{qr} + \frac{d}{dt}(\theta_r)\phi_{ds} + \frac{d}{dt}(\phi_{qr}) \tag{6.45}$$

Flux linkages:

$$\phi_{ds} = L_{ss} i_{ds} + L_m i_{dr} \tag{6.46}$$

$$\phi_{qs} = L_{ss} i_{qs} + L_m i_{qr} \tag{6.47}$$

$$\phi_{dr} = L_m i_{ds} + L_{rr} i_{dr} \tag{6.48}$$

$$\phi_{qr} = L_m i_{qs} + L_{rr} i_{qr} \tag{6.49}$$

where

$$L_{ss} = L_s + L_m ,$$

$$L_{rr} = L_r + L_m ,$$

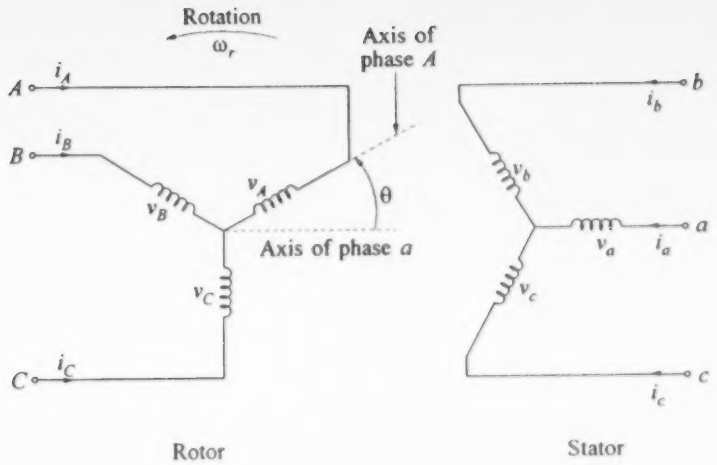


Fig. 6.13 Induction Motor

L_s, L_r = stator and rotor leakage inductances, respectively,

L_m = mutual inductance, and

ϕ = the flux linkage.

All the quantities are in per unit. Eliminating the rotor currents, we have from (6.48),

$$i_{dr} = \frac{\phi_{dr} - L_m i_{ds}}{L_{rr}} \tag{6.50}$$

Substituting (6.50) into (6.46), we have

$$\phi_{ds} = L_{ss} i_{ds} + \frac{L_m (\phi_{dr} - L_m i_{ds})}{L_{rr}}$$

$$= \frac{L_m}{L_{rr}} \phi_{dr} + \left(L_{ss} - \frac{L_m^2}{L_{rr}} \right) i_{ds} \tag{6.52}$$

Substituting (6.52) into (6.43), we have

$$v_{qs} = R_s i_{qs} + \omega_s \frac{L_m}{L_{rr}} \phi_{dr} + \omega_s \left(L_{ss} - \frac{L2_m}{L_{rr}} \right) i_{ds}$$

$$= R_s i_{qs} + x_s^{'} i_{ds} + E_q^{'} \tag{6.53}$$

where

$$E_q^{'} = \omega_s \frac{L_m}{L_{rr}} \phi_{dr},$$

$$x_s^{'} = \omega_s \left(L_{ss} - \frac{L_m^2}{L_{rr}} \right), \text{ transient reactance of the induction machine.}$$

Similarly, for the d –axis,

$$v_{ds} = R_s i_{ds} - x_s^{'} i_{qs} + E_d^{'} \tag{6.54}$$

where

$$E_d^{'} = -\omega_s \frac{L_m}{L_{rr}} \phi_{qr}.$$

From (6.48) and (6.49), we derive the differential equations for $E_d^{'}$ and $E_q^{'}$.

$$\frac{dE_q^{'}}{dt} = \omega_s \frac{L_m}{L_r} \frac{d}{dt} (\phi_{dr})$$

$$= \omega_s \frac{L_m}{L_{rr}} \left[-R_r i_{dr} + \frac{d}{dt} (\theta_r) \phi_{qr} \right]$$

$$= -\omega_s \frac{L_m}{L_{rr}}\left[\frac{\phi_{dr} - L_m i_{ds}}{L_{rr}}\right] + \omega_s \frac{L_m}{L_{rr}} \frac{d}{dt}(\theta_r)\phi_{qr}$$

$$= -\frac{1}{\tau_o'}\left[E_q' - \omega_s \frac{L_m^2}{L_{rr}} i_{ds}\right] + \omega_s \frac{L_m}{L_{rr}} \frac{d}{dt}(\theta_r)\phi_{qr}$$

$$= -\frac{1}{\tau_o'}\left[E_q' - (x_s - x_s')i_{ds}\right] - E_d' \frac{d}{dt}(\theta_r) \qquad (6.55)$$

where

$$(x_s - x_s') = \omega_s \frac{L_m^2}{L_{rr}}.$$

Similarly,

$$\frac{dE_d'}{dt} = -\omega_s \frac{L_m}{L_r} \frac{d}{dt}(\phi_{qr})$$

$$= -\frac{1}{\tau_o'}\left[E_d' + (x_s - x_s')i_{qs}\right] + E_q' \frac{d}{dt}(\theta_r) \qquad (6.56)$$

where

$$\tau_o' = \frac{L_{rr}}{R_r},$$

$x_s = \omega_s L_{ss}$, and

$$\frac{d}{dt}(\theta_r) = \frac{\omega_s - \omega_r}{\omega_s}.$$

ω_s is the angular velocity of the rotor in radian per second. θ_r is the angle by which the d – axis leads phase A – axis of the rotor. If the rotor slip is s, the d – axis is advancing with respect to a point on the rotor at the rate

$$\frac{d}{dt}(\theta_r) = s \cdot \omega_s \tag{6.57}$$

The rotor acceleration equation, with time expressed in seconds is

$$\frac{d^2}{dt^2}(\theta_r) = \frac{1}{2H}(T_e - T_m) \tag{6.58}$$

where the electromagnetic torque is given by

$$T_e = E'_d i_{ds} + E'_q i_{qs} \tag{6.59}$$

and, the load torque can be expressed as,

$$T_m = T_0(\omega_r)^m \tag{6.60}$$

or

$$T_m = T_0\left(a\omega_r^2 + b\omega_r + c\right) \tag{6.61}$$

depending on the type of loads where $\omega_r = \frac{d}{dt}(\theta_r)$. Figure 6.14 shows the block diagram of the induction motor that is used in TSSP.

Load modeling itself is a very difficult task as meaningful data are simply unavailable and collecting and processing load data is expensive. Moreover, exact simulation of loads requires that at each time step, a load flow study be carried out in order to obtain the accurate bus voltage. This will increase the computation effort beyond an acceptable level. However, an alternative is to calculate a new operating point by a fast decoupled load flow program[6–7] only at the instant when an event or a disturbance happens. At other times, the bus voltage is held constant in load representations as computed in the latest load flow study.

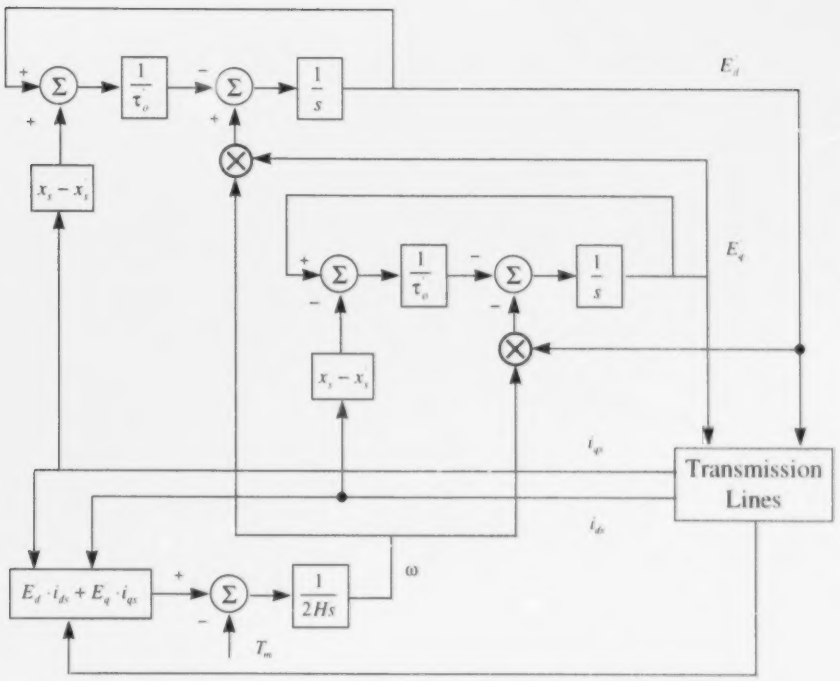


Fig. 6.14 Block Diagram of An Induction Motor

6.2.7 DC Transmission

An alternating current transmission system is represented by a positive–sequence circuit in transient stability simulations. In order to achieve high performance in power systems, direct current transmission system is used. It provides maximum flexibility in power control and enhances transient stability.

Figure 6.15 shows a schematic diagram of a direct current(DC) link between two AC transmission systems. The transformers provide an ungrounded three–phase voltage source of appropriate level to the valve bridge of the rectifier/inverter. The DC line is similar to an AC line, except it has different numbers of conductors and spacing between conductors. Figure 6.16 is the corresponding equivalent circuit. The direct current flowing from the rectifier to the inverter is given by

$$I_d = \frac{V_{dor}\cos(\alpha) - V_{doi}\cos(\gamma)}{R_{cr} + R_L - R_{ci}} \tag{6.62}$$

where

α = the ignition delay angle,

γ = the excitation advance angle.

The power at the rectifier/inverter terminals is given by, respectively,

$$P_{dr} = V_{dr} + I_d \tag{6.63}$$

and

$$P_{di} = V_{di} I_d = P_{dr} - R_L I_d^2 \tag{6.64}$$

Figure 6.17 shows a schematic diagram of a converter. Let B be the number of bridges in series, T the transformer ratio. Then the ideal no–load voltage on the DC side is given by

$$V_{do} = \frac{3\sqrt{2}}{\pi} BTV_t \tag{6.65}$$

where V_t is the terminal voltage at the AC side.

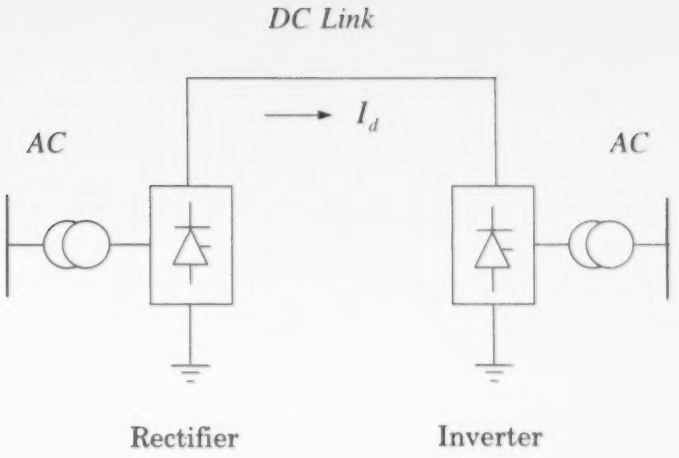


Fig. 6.15 Schematic Diagram of A DC Line

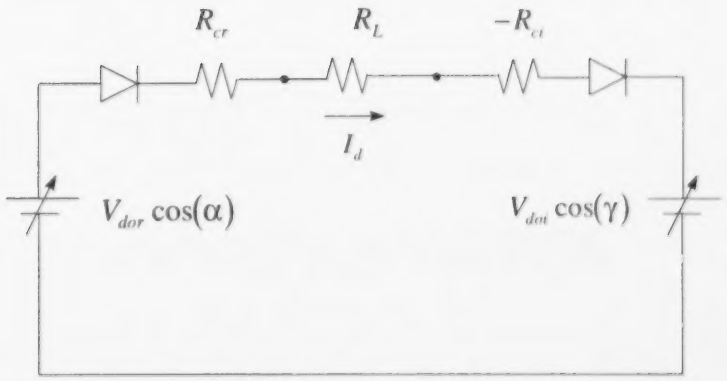


Fig. 6.16 Equivalent Circuit of A HVDC Link

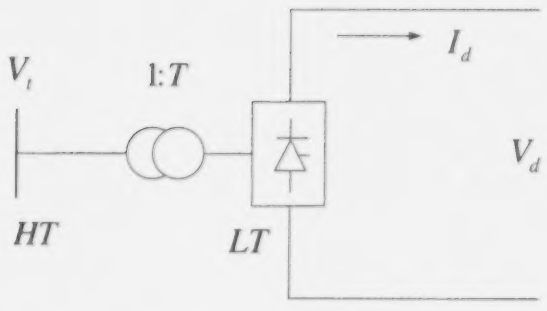


Fig. 6.17 Converter Representation

When there is load, there will be voltage drop across each bridge. Now the voltage on the DC side is expressed as one of the following two

$$V_d = V_{do}\cos(\alpha) - I_d B\left(\frac{3}{\pi}x_c\right) \tag{6.66}$$

$$V_d = V_{do}\cos(\gamma) - I_d B\left(\frac{3}{\pi}x_c\right) \tag{6.67}$$

where

$R_c = \frac{3}{\pi}wL_c$, the equivalent commutating resistance.

The power factor can be expressed as

$$\varsigma = \cos^{-1}\left(\frac{V_d}{V_{do}}\right) \tag{6.68}$$

$$P = V_d I_d \tag{6.69}$$

$$Q = P\tan(\varsigma) \tag{6.70}$$

where P, Q are active and reactive power, respectively.

For a more detailed representation of DC transmission system and its control for transient stability study, the reader is referred to [6–10]. This subject is further treated in Chapter 7 where TSSP in SIMULINK is presented.

6.3 System Disturbances

The modeling of system disturbances is an important aspect in simulating the dynamic behavior of a power system with acceptable accuracy. Both symmetrical and asymmetrical disturbances should be modeled. A symmetrical disturbance can be easily modeled as the system will remain symmetrical under such a disturbance. Under asymmetrical disturbances, the solution of the stability problem for the disturbed three–phase network can be carried out by a three–phase stability program, where the symmetrical component method is employed to resolve the system into three symmetrical three–phase systems. Therefore, the total computations will increase up to three times. On the other hand, it has been found [6–15] that the average braking torque produced by the reaction of the two magnetic fields (one produced by the negative sequence current and the other by the rotor winding current) is approximately zero, and that zero sequence currents yield a zero component torque as the three–phase zero sequence currents are electrically in phase and have 120 degree displacement in space. Therefore,

only the positive sequence quantities are needed to be taken into account during a transient process. However, the negative and zero sequence networks have to be incorporated into the positive sequence network at the fault point.

Each disturbance can be described and modeled as a sequence of events. They may include pre–fault, fault, post–fault and line restoration, i.e., autoreclosure if any, etc. The admittance matrix of the network is modified each time there is a change in the configuration of the network during the disturbance. As most faults in a practical power system are asymmetrical in nature, it is important to know what behavior a system would exhibit under such disturbances and what measures should be taken to maintain system stability while maximizing the availability of power supply to customers. Detailed modeling aspects will be further covered in Chapter 7 when presenting TSSP in SIMULINK environment.

CHAPTER 7

A TRANSIENT STABILITY SIMULATION PACKAGE (TSSP) UNDER SIMULINK

This chapter describes the Transient Stability Simulation Package (TSSP) under SIMULINK environment. It introduces the package under DOS operating system. It then points out the areas in TSSP where improvement can be made in order to better the performance of the package. For example, handling generator saliency is difficult in stability simulation studies. This difficulty will be overcome in the new software to be developed. Building blocks of various power apparatuses are developed and a rich library including such blocks and MATLAB functions is formed. Basic instructions on how to construct a simulation model for a power system are given, including an illustrative example.

7.1 TSSP UNDER DOS

Time domain simulation of power systems has been used for stability studies for decades[7–1]~[7–4]. Various large scale simulation packages such as the EPRI–ETMST (Extended Transient/Mid Term Stability Program) and the PTI–PSS/E (Power System Simulation/E Program) are in wide use. A number of educational softwares are also reported in the literature [7–5]~[7–8].

Every simulation package has its own advantages and shortcomings. Most packages do not have all the features and flexibility that one needs for either education or research purposes. The possibility of changing the capability of the package is often limited if not impossible. There has been an urgent need to have a package that has the capability of fulfilling most requirements in both teaching and research. To answer this need, a transient stability simulation package(TSSP) implemented on an IBM PC has been developed[7–9]. This package has been utilised to perform stability studies of the WSCC system [7–10] and the New England Test system[7–11]. As a research tool, the package has also been used to conduct a stability investigation of a longitudinal power system having 69 buses and 12–machines [7–12]. The following is a list of tasks that can be performed using the TSSP:

- Machine modelling of varying complexity;
- Transient simulation of induction motors;
- Modelling of symmetrical and asymmetrical disturbances;
- Various load modelling;
- Modelling of AVR, AGC and PSS;
- Interfacing of user–defined modules;
- Load flow studies; and
- Short circuit calculations.

The transient behaviour of a power system component or a controller is modelled by a set of first order differential equations. Based on the implicit integration method [7–13], this set of differential equations can be simulated by building a block diagram with integrators and DC gains. Each block diagram forms an independent module to be called upon by the co–ordinator in the TSSP. This package has modules for four different synchronous machine models, one induction motor model, all the current IEEE recommended excitation system models, the standard power system stabiliser model[7–14], and speed governing and turbine system models[7–15]. Different load models can also be included [7–16].

A maximum of three input data files are needed to perform a transient stability simulation. Each data file consists of a number of data blocks. In each block, data items are entered as a record in free format with a comma between items. A dash line (/) signifies the end of a record. Preceding each block, there are comments and definitions specifying what data are to be entered and how. Any length of comments can be added into the files as long as an exclamation mark (!) precedes each comment line.

Data file *pqlf.inp* contains all the information required to perform a load flow study. It also provides the freedom to use different system base MVA, per unit or nominal unit system. Data file *ntwk.inp* contains fault information, sequence network parameters and branch incidence of the zero sequence network. Data file *mach.inp* contains parameters for machine,

excitation system (AVR), speed governor and turbine system (AGC) and power system stabiliser (PSS) in four blocks. Each controller is identified in the corresponding block by its machine number. If a controller is absent from a machine, a record 888/ is entered to signify this fact.

Three sample data files are provided in TSSP. The user can create his/her own data files by copying and modifying them with a DOS editor or any word processor according to the system under study. Any error in the data files will be automatically detected and located when the programs are started. The format of the data files is designed to provide maximum flexibility and minimize frustration in data input. This design was inspired after experiencing the "sophisticated" and "non user–friendly" data input in EMTP [7–17].

The overall structure of the package is shown in Figure 7.1. To initialise the TSSP, the fast decoupled load flow (FDLF) is run first. Then the short circuit calculation (STCC) program is executed. The results of the two runs are passed onto the TSSP co–ordinator internally. Finally the transient simulation program (TSSP) is run. The end result of a simulation is stored in the file *simu.out*. A data sorting and plotting program (WPLOT) can be employed to plot a maximum of five curves on one screen and up to fifteen variables can be sorted out at a time.

7.2 NECESSITY OF IMPROVEMENT

TSSP is a numerical realisation of solving a large mathematical system composed of both differential and algebraic equations. Let x denote the system state variable, and y the auxiliary variable. Then the mathematical system describing the dynamic behaviour of a power system can be expressed as

$$\frac{d}{dt}(x) = f(x, y) \tag{7.1}$$

$$0 = g(x, y) \tag{7.2}$$

where

x = the state variables, and

y = the auxiliary variable.

A numerical solution to the system can be obtained by different algorithms, assuming that an initial condition is known. This initial condition is obtained by a load flow program. The solution procedure in TSSP consists of the following steps;

1. With the initial values $x^{(0)}, y^{(0)}$ known, and with a given disturbance in the system, a new value of the auxiliary variable $y^{(1)}$ is calculated from (7.2), keeping the state variable at $x^{(0)}$.
2. The derivative of the state variable is calculated from (7.1).
3. The new value of the state variable $x^{(1)}$ is calculated using a numerical method such as the implicit integration.

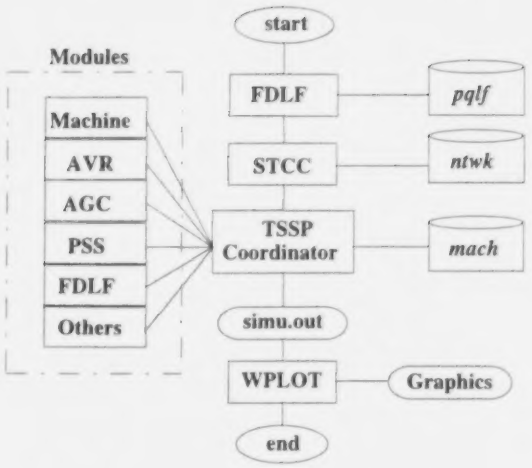


Fig. 7.1 Overall Structure of the TSSP

This calculation process is continued until the simulation time is reached. At each time step, the calculated values of $x^{(i)}$ and $y^{(i)}$ are stored so that they can be plotted at the end of simulation.

Because of the fact that there exists saliency in synchronous machines, the solution of (7.2) is accomplished by an iterative procedure in TSSP[7–2]. Depending on the structure of the saliency, this iterative procedure may lead to divergence and no solution can be obtained at certain times and with certain network configurations. For real time studies, this iterative process may impose unacceptable computation effort. Another disadvantage of this package is that to add any new features to the program, such as displaying a variable which is not listed in the plotting utility, modelling a new disturbance, or interfacing a user–defined model, part of the package has to

be recompiled. This compilation process has to be continued until the desirable results are produced.

7.3 TSSP UNDER SIMULINK

7.3.1 Introduction

SIMULINK is an interactive block–diagram environment for modeling, analyzing, and simulating dynamics of physical and mathematical systems. It provides point–and–click interface for all model building. Block–diagram and differential equation models, continuous–time, discrete–time, hybrid models, linear and nonlinear models, trained neural networks and fuzzy logic models, can all be simulated in this environment.

Though SIMULINK is powerful in simulating the above systems, it can not be directly applied to power system simulations. The reason is that power systems run a three–phase complex number system and their control systems are modeled by real number transfer functions. In order to comply with the rules of SIMULINK, a power system has to be decoupled from complex representation to real number representations which are asymmetrical when salient synchronous machines are present in the system. In the section that follows, the basics of SIMULINK are discussed. Then detailed building blocks for the modules in TSSP will be constructed using SIMULINK library and MATLAB functions, forming the TSSP Library.

7.3.2 Basics of SIMULINK

To simulate a dynamic system, we need to create its model in *a block diagram* window using mouse driven commands. As SIMULINK uses the metaphor of a block diagram to represent dynamic systems, creating a model is much like drawing a signal flow graph. The individual blocks need not be drawn, but are represented by input/out blocks from the SIMULINK standard library. Once the model is defined, its dynamic performance can be simulated by built–in analysis tools. The progress of the simulation can be viewed while the simulation is running, and the final results can be made available in the MATLAB workplace when the simulation is complete. Specifically, the following steps can be followed to create and analyse the model for a given dynamic system.

1. At MATLAB prompt, type `simulink` and press return. This will display the standard library of block diagram.
2. Pull down the MATLAB file menu and open up a new model file which is a blank *block diagram* window.
3. Copy individual blocks from the standard library to your *block diagram* window; and repeat this step until the model is complete.
4. Save that model you just defined as a *m*–file which you can modify or run later on.

7.3.3 Building Blocks

(1) *Synchronous Generators*

In Chapter 6 various synchronous generator models are presented. In this section, their simulation models will be built. This is done by showing one detailed example, i.e., building of the *two–axes model with subtransient*. The other three models will be listed in Appendix B. Figure 7.2 shows the simulation model of a synchronous generator. A functional block for both an excitation system and a speed governing and turbine system is included. These functional blocks have to be replaced by the simulation models of their corresponding physical equipment.

The input to the model is a vector output from a MATLAB function that solves the network equations of the power system under study. It can be expressed as follows, for generator number 1:

$$in_1 = \left[i_d, i_q, V_x, V_y, i_x, i_y \right]^{\mathrm{T}} \tag{7.3}$$

The output of the model is a vector that can be expressed as follows:

$$out_1 = \left[E_q^{'}, E_q^{''}, E_d^{''}, \delta \right]^{\mathrm{T}} \tag{7.4}$$

The output of the excitation system is E_{fd}. And the output of the speed governing and turbine system is P_m. The rest of the model is to simulate the differential equations (6.18) – (6.22). Note that the memory blocks(M01–M03) in the model function as a breakdown of a complete loop for simulation purpose. Values of parameters are passed to their corresponding variables in

the various blocks from the workplace where these variables are initialized before simulation is started.

(2) *Excitation Systems*

As an example to show how to build a simulation model for an excitation system, we choose the DC Type 1 excitation system. Figure 6.3 shows its block diagram. Figure 7.3 shows a simulation model of the system. The input to the model has two variables, i.e., the terminal voltage and the output signal from a power system stabiliser. The output of an excitation system is always the excitation field voltage, E_{fd}. There are two MATLAB functions; i.e., VT block converts a voltage in complex components to its magnitude and SE calculates the saturation factor of the excitation curve at an operating point. Other types of excitation systems are given in the TSSP library.

(3) *Prime Movers*

The hydro turbine and governor system, shown in Figure 6.5, can be accurately simulated in TSSP, since nonlinear elements of a dynamic system can be easily modelled in SIMULINK. Figure 7.4 shows the simulation model in which there are two subsystems. Each of these subsystems simulates a transfer function with initial variable values. The input to the model is the machine speed in per unit and the output is the mechanical power in per unit exerted to the shaft of generating system.

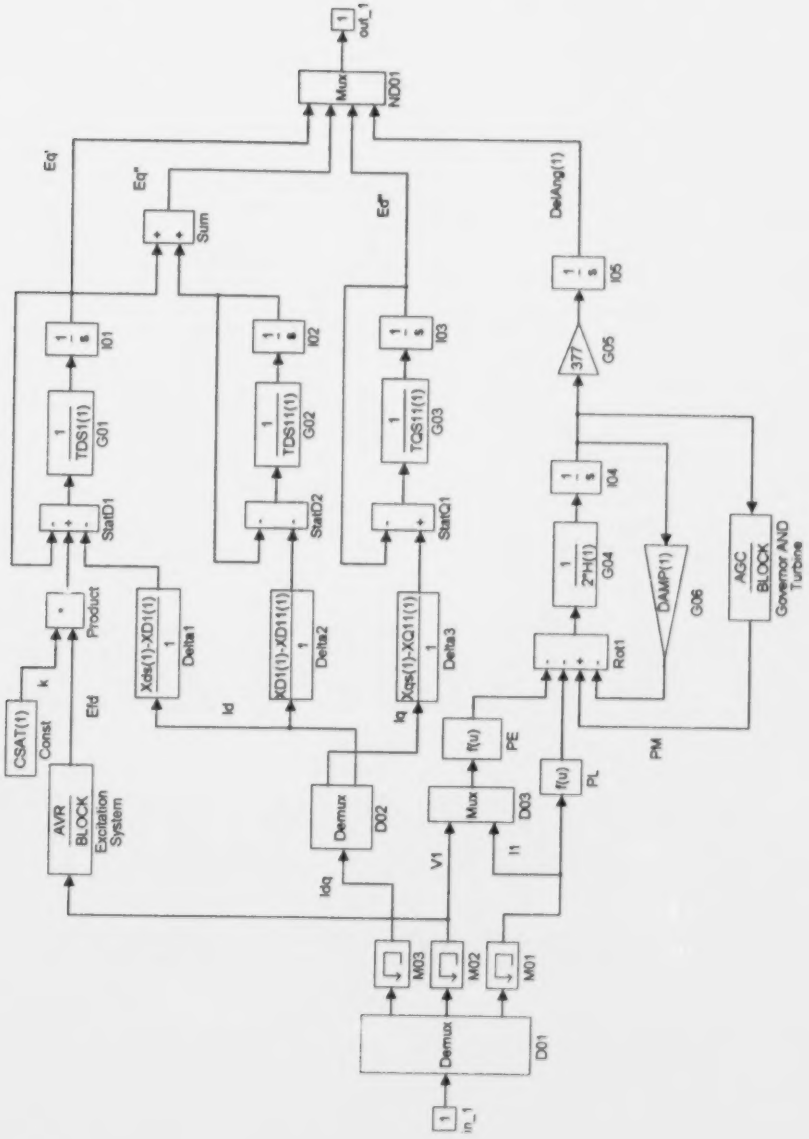


Fig. 7.2 Simulation Model of A Synchronous Generator

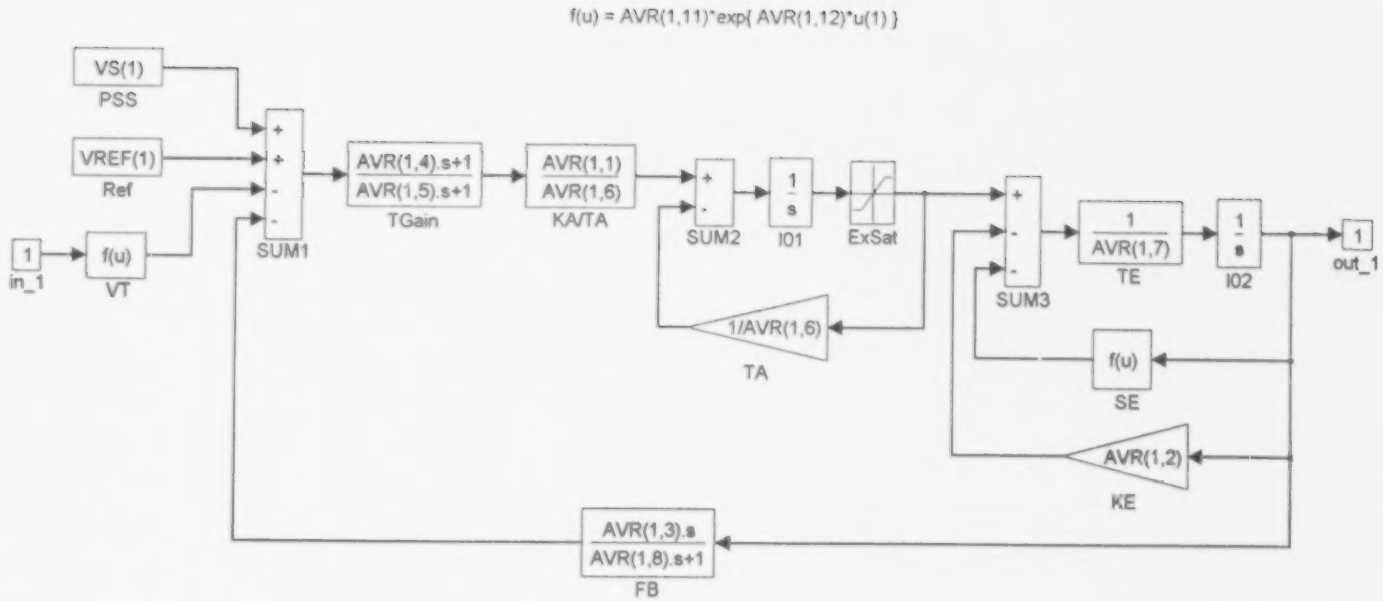


Fig. 7.3 Simulation Model of the DC Type 1 Excitation System

The steam turbine system, shown in Figures 6.7 and 6.8, is simulated by the model shown in Figures 7.5 and 7.6. The input to the governor model is the incremental change of the shaft speed of the generator. The output is the valve positioning that controls the inlet of steam and is equivalent to the total power input to the turbine which converts the heat energy to mechanical energy and outputs it as P_m. Other steam turbines are also included in the TSSP library.

(4) *Conventional Power System Stabilisers*

Three IEEE conventional power system stabilisers(PSSs) are presented in Chapter 6. Their simulation models in SIMULINK are constructed here. They are also included in the TSSP library. The input to a PSS can be a combination of different signals, as has been pointed out in Chapter 6. Figures 7.7, 7.8 and 7.9 give the details of these simulation models.

(5) *Induction Motors*

The block diagram of an induction motor is shown in Figure 6.13. As an induction motor is treated like a generator in TSSP, its simulation model in SIMULINK is similar to that of a generator too, as shown in Figure 7.10. The input to the model is a vector output from the network solution MATLAB function, as expressed in (7.3). The output of the model is a vector given by

$$out_1 = \left[E_q', E_d', \delta\right]^T \tag{7.5}$$

The load of the induction motor is passed to the model from the workplace by the variable TM(1) which is calculated by either (6.60) or (6.61). There are two MATLAB functions; PE calculates the electric power the motor consumes and PL calculates the power loss in the stator windings. The rest of the model simulates the differential equations (6.5) – (6.59) of the system.

(6) *DC Transmission Lines*

A three–phase alternating current(AC) transmission line system is represented by the positive–sequence network of one of the three phases in transient stability studies(see next section for more detail). DC transmission lines can be simulated by three categories of models; (a) simple model, (b) response model, and (c) detailed model. Depending on its application, each DC transmission line is designed to meet specific requirements in the overall performance of the system, therefore, no standard model is developed to represent DC links in stability studies[7–18].

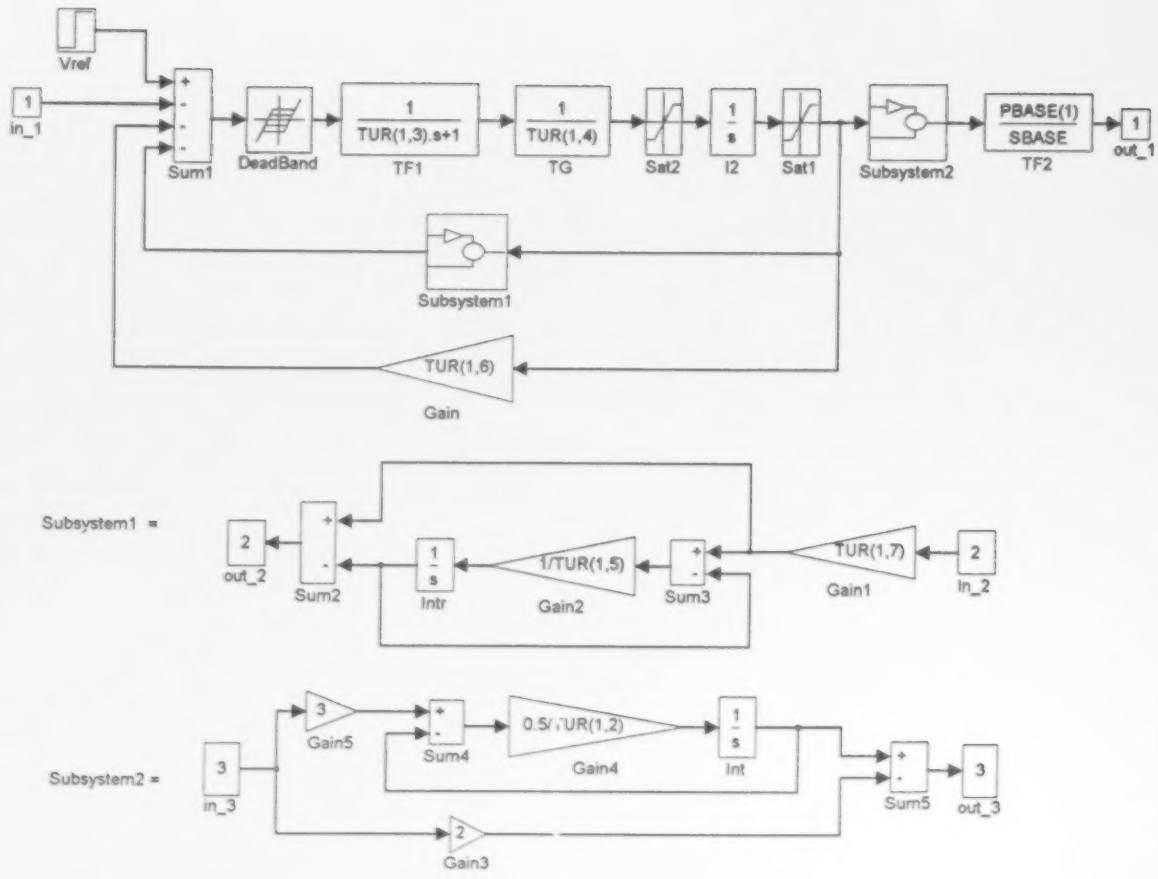


Fig. 7.4 Simulation Model of the Hydrogovernor Turbine System

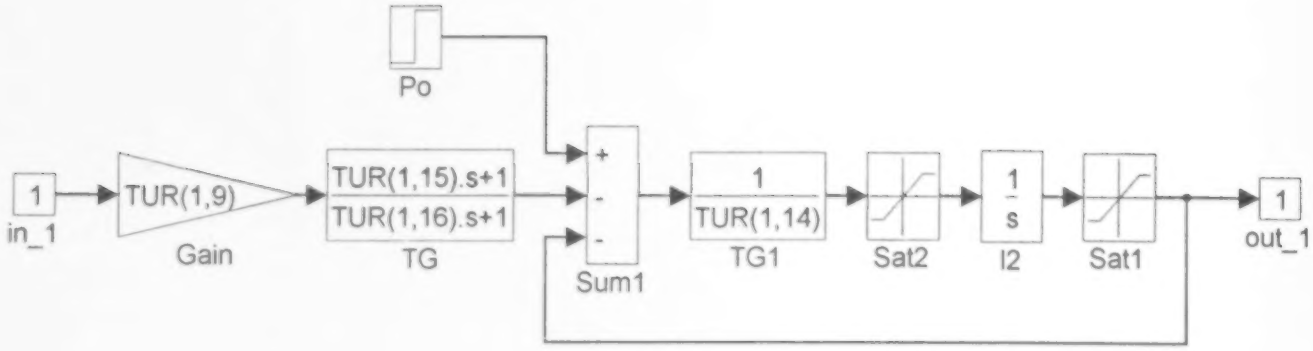


Fig. 7.5 Speed Governing System for Steam Turbines

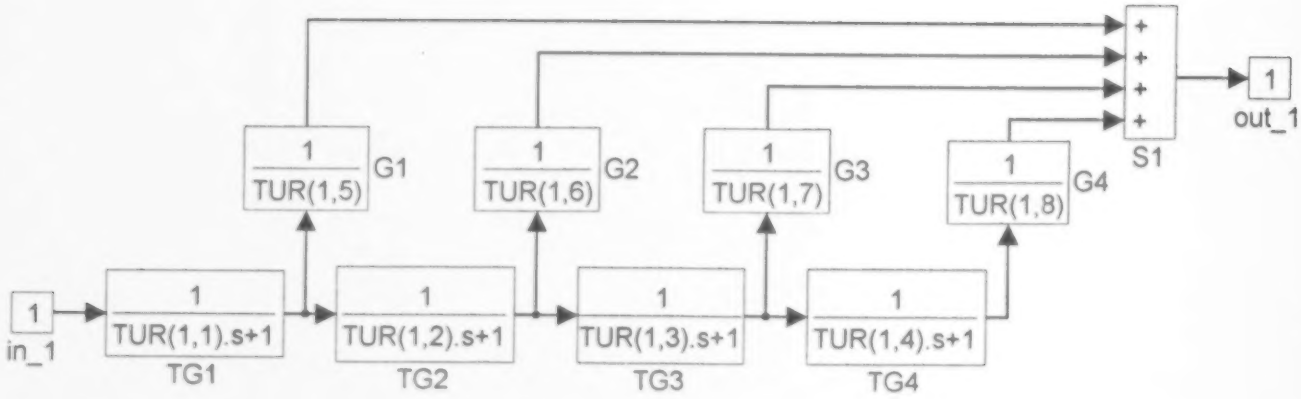


Fig. 7.6 Tandem–Compound Double–Reheat Steam Turbine

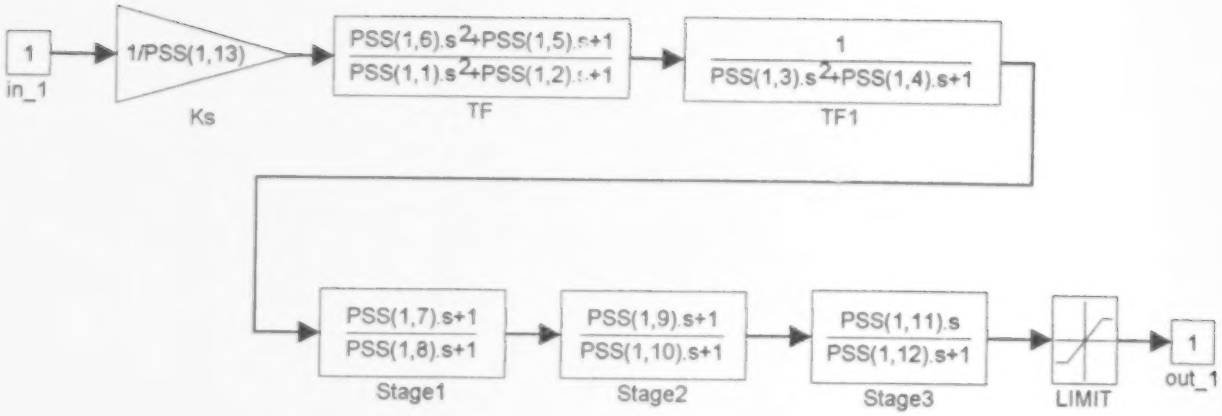


Fig. 7.7 Simulation Model of IEEE Standard PSS

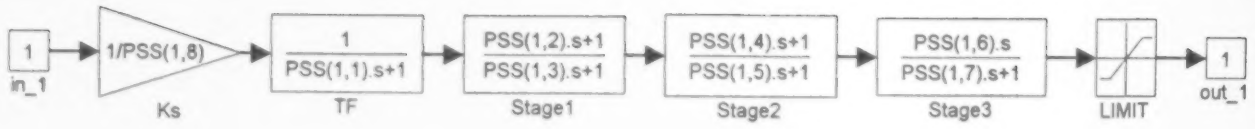


Fig. 7.8 Simulation Model of IEEEST PSS

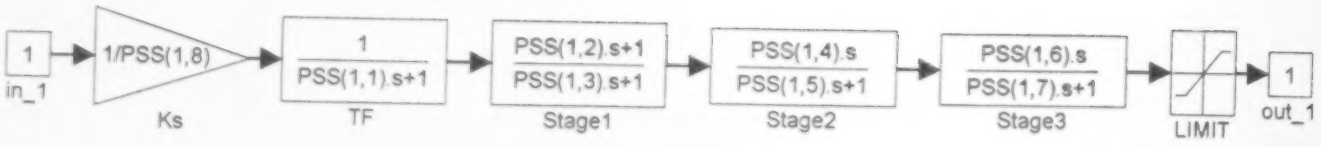


Fig. 7.9 Simulation Model of IEEESN PSS

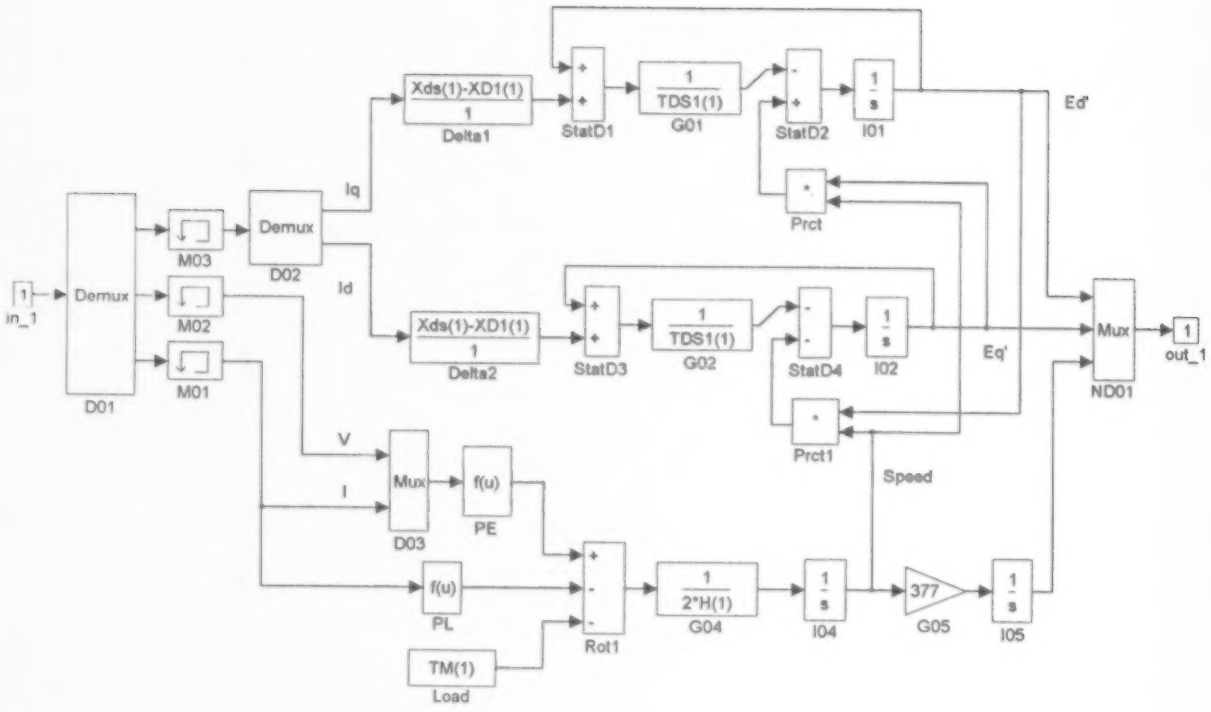


Fig. 7.10 Simulation Model of An Induction Motor Model

(7) *Network Function*

In a power system, generators, loads and other power apparatuses are connected though transmission lines to form a composite dynamic system. As generators and their control systems are modelled by real number transfer functions while the network is represented by complex numbers, we need to decouple the complex number system. This is done in TSSP through a MATLAB function in which an AC system is modelled by

$$\begin{pmatrix} I_{xy1} \\ I_{xy2} \\ . \\ I_{xyn} \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} & . & Y_{1n} \\ Y_{21} & Y_{22} & . & Y_{2n} \\ . & . & . & . \\ Y_{n1} & Y_{n2} & . & Y_{nn} \end{pmatrix} \begin{pmatrix} V_{xy1} \\ V_{xy2} \\ . \\ V_{xyn} \end{pmatrix} \tag{7.6}$$

where $I_{xyi} = \begin{bmatrix} I_{xi} & I_{yi} \end{bmatrix}^{\mathrm{T}}$, the injected current,

$V_{xyi} = \begin{bmatrix} V_{xi} & V_{yi} \end{bmatrix}^{\mathrm{T}}$, the voltage at bus i, and

$Y_{ij} = \begin{pmatrix} G_{ij} & -B_{ij} \\ B_{ij} & G_{ij} \end{pmatrix}$, its elements being the real and imaginary parts of the element $\left(G_{ij} + jB_{ij}\right)$ at position (i, j) of the admittance matrix of the network.

The interfacing point between the network and generators, loads and other power apparatuses is the bus terminal where injected current and bus voltage hold true for both parties. With no approximation and no iteration, this interfacing is accomplished in the MATLAB function, using exact algebraic equations as derived in Chapter 6. For any generator or load model,

if its interfacing condition with the network is expressed in terms of voltage and current in $d-q-0$ quantities, they can be transformed into quantities in $x-y$ coordinates. For example, a current transformation is accomplished by

$$\begin{pmatrix} I_x \\ I_y \end{pmatrix} = \begin{pmatrix} \cos(\delta) & \sin(\delta) \\ \sin(\delta) & -\cos(\delta) \end{pmatrix} \begin{pmatrix} I_q \\ I_d \end{pmatrix} \tag{7.7}$$

where δ is the angle between the reference $x-$axis in $x-y$ coordinates and the $q-$axis of the device to be modelled. This transform applies to voltage too. Detailed information on the network function is given in Appendix B. This topic is further covered in the coming section where model construction for a power system is described.

7.3.4 Model Construction

In this section, we demonstrate how to construct a simulation model for a simple one–machine infinite bus power system. The procedure can be extended to the case of multimachine systems which will be covered in Chapter 8. Figure 7.11 shows a single–line diagram of the system. Assume that the generator can be simulated by the *two–axes with transient* model(GEN_3). The excitation system is simulated by ST_4. The speed governing and turbine system is simulated by GOV_221. The load is an induction motor modelled by IndMotor block from the TSSP library. The infinite bus is simulated by GEN_0 block. Figure 7.12 illustrates the final simulation model of the power system. Note that the MATLAB function is a

m-file referred as find_vif.m, meaning to find voltage(v) and current(i). This file is listed in Appendix B. The input vector to a MATLAB function can have only a single argument with a number of elements. This allows access to the numerous variables output from various devices each of which may have an arbitrary number of outputs. The output of the function is also a vector with the number of components equal to a multiple of the number of the devices modeled. This is necessary because the elements of the output vector have to be accessed by each device appropriately. A simulation function referred to as Demux can divide its input only into equal parts.

7.4 TSSP's Potential with Simulink

We have seen some merits of TSSP under SIMULINK environment, and other potential applications is of interest. With the new DSP Blockset and a real-time C code generator[7-19], we can develop real-time software for target hardware and test the generated C code in actual applications. This C code generator is useful for batch file simulation studies too, as the size of a power system increases, model simulation in SIMULINK may be slow. The generated portable C code can be used in different situations. Another advantage is that we can use advanced control techniques such as ANNs and FL to design intelligent control systems. With TSSP in SIMULINK, we can prototype, simulate and implement a control on a single personal computer.

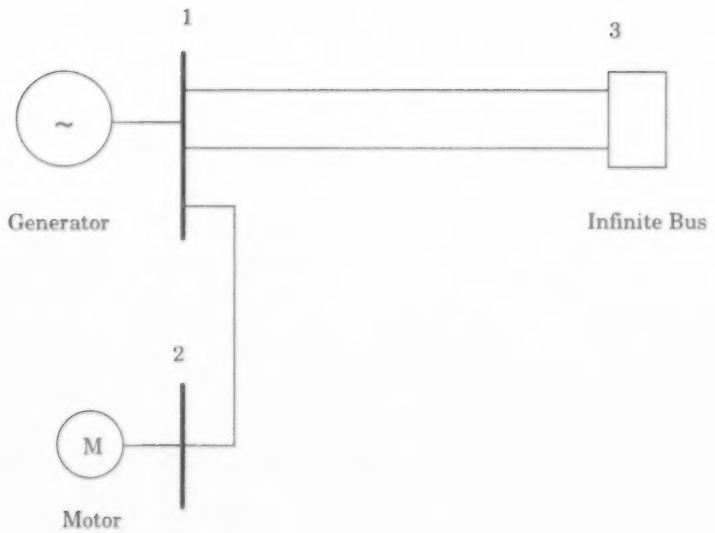


Fig. 7.11 Single–Line Diagram of the One–Machine System

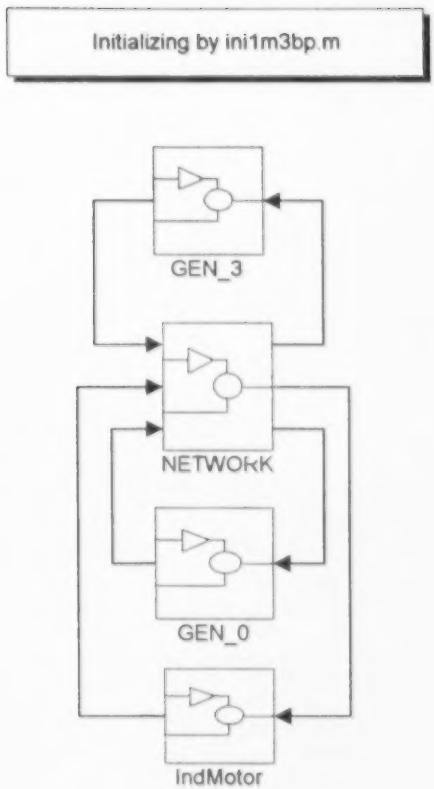


Fig. 7.12 Simulation Model of the One-Machine System

CHAPTER 8

STABILITY ENHANCEMENT OF A MULTIMACHINE POWER SYSTEM USING AN ANN-FL PSS

In a stable power system, all the synchronous generators operate in synchronism. When perturbed, they must either return to their original state if no net change of power occurs or reach a new state asymptotically without losing synchronism. It is difficult to theoretically prove whether a multimachine power system is stable. The objective of this chapter is to show how power system stability can be further enhanced by using an ANN–FL power system stabilizer(PSS). Two power system stabilizers based on the ANN–FL modelling technique, one using speed deviation as input and the other using accelerating power, are designed and applied to a two–machine infinite–bus system to evaluate their performance.

8.1 INTRODUCTION

To meet a steady demand growth of interconnected power systems, the size of generating units has been increased, and high–speed excitation systems have been introduced. By doing so negative damping to the interconnected system is resulted, causing low–frequency oscillations. Weak line transmissions are

another contributing factor to these oscillations. Nevertheless, electro–mechanical oscillations of low frequency become inherent characteristics of power systems. The presence of such oscillations has been reported worldwide[8–1]~[8–5]. Low frequency oscillations are also called system modes. Principally, there are two categories of modes: inter–area modes associated with one group of generators or plants at one end of a tie–line oscillating against another group at the other end and local modes associated with weakly connected power systems or remote generating units weakly connected to a large power system. Different system modes can occur simultaneously. This coupling among modes makes it difficult to define 'cause and effect' relationships in analysing the dynamic behaviour of a multi–machine system, especially for inter–tie line oscillations [8–6]~[8–9]. Eigenvalue analysis is a fundamental technique to study the nature of system modes of a power system [8–10,8–11]. A positive real part of a swing mode indicates a negatively damped mode which needs controlling to ensure stable system operation. This technique is also used in this thesis in designing the conventional PSSs for the generators.

The common remedy for inadequate damping is to utilize additional excitation control by means of power system stabilizers [8–7,8–12]. After more than two decades of research and practical applications, PSSs are now in wide use in power systems. In most PSS applications, only local feedback control is used. Multivariable and optimal stabilizers can also be

theoretically designed but may not be implemented because of the difficulties in accessing most of the feedback variables. Coordination in PSS design has been taken into account by either eigenanalysis [8–13], or frequency domain methods [8–14] or a hybrid of the two [8–15]. No matter what algorithm is employed in tuning the settings of a PSS, the more information dependent on operating conditions is used, the less robust the PSS will be to system changes.

In [8–5], a simple PSS design procedure is proposed and used in a practical power system of longitudinal structure. Usually, system modes must be known in PSS design. It is also generally true that the damping (real part) of a system mode is more sensitive to operating conditions than the frequency (imaginary part) of that mode. Reduction of these sensitivities of a mode increases the robustness of the PSS designed based on this mode. Instead of using a specific frequency for a particular PSS design, an average frequency is derived for each coherent group of generators that oscillate together. The coherent generation groups can be identified by available methods [8–16,8–17]. Another parameter needed in tuning a PSS is a lead/lag time constant spread [8–7,8–8]. Nonlinear simulation is utilized to determine the optimum gain of each PSS. Input signals to all the PSSs in each coherent generation group may be communicated with each other among the strongly coupled generators. The total coupling factors [8–18] computed from eigenanalysis can be employed to take into account these

interactions among generators. The design procedure developed in this paper was successfully used in another research project leading to the publication of [8–19]. This procedure will be used in this thesis to design benchmark PSSs for training ANN–FL power system stabilizers.

Though, there have been many examples of successful applications of conventional PSSs, difficulties include the tuning of a PSS for a wide range of operating conditions, as well improving the performance under different contingencies. PSSs that utilize shaft speed as input signal may result in torsional oscillations. PSSs using power signals may cause voltage fluctuations. In recent years, alternative control strategies have been investigated, especially intelligent control. Chapter 1 has reviewed a few intelligent control schemes based on artificial neural networks(ANNs) and fuzzy logic control(FLC). In this category of applications, publications [8–20]~[8–25] are closely related to the subject of this chapter. The theme of those papers is to develop an event driven controller representing a nonlinear mapping from the input to the output variables. Fuzzy logic control is a structured representation technique, it does not allow learning nor adaptation. On the other hand, artificial neural networks are unstructured numerical estimators that can learn, generalize and process massive sensory information. An intelligent controller designed on the basis of the hybrid theory of ANN and FL presented in Chapter 4 combines the merits of both modelling techniques.

8.2 STABILITY ANALYSIS

8.2.1 A Multimachine Power System

A single–line diagram of the multimachine power system used in the simulation studies is shown in Figure 8.1. The system consists of two hydro generators, one load, and one infinite bus. Generator #1 is a test machine, modeled by a fifth–order model with subtransient (GEN_4)(see Appendix D) simulating a single generator in the system. Generator #2 is an equivalent machine modeling a local power system. It is simulated by a fifth–order model with transient (GEN_3). The infinite bus (GEN_0) is the connecting point with a remote power system. Each machine is equipped with a simple static (SCR) excitation system (ST_4) and a standard hydro speed–governing and turbine system (GOV_221). Figure 8.1 shows the parameters of the network. Parameter values for the various power devices are listed in Tables C.1–C.3.

Two load flow studies were carried out using the decoupled fast load flow program(Chapter 7). Tables 8.1–8.2 list the obtained results, including voltage magnitude and angle, injected real and reactive power, power flow in each transmission line and power losses. The operating point of Case A will be used for conducting eigenanalysis and later on for designing conventional power system stabilizers(CPSSs). The operating point of Case B is designed for performance evaluation.

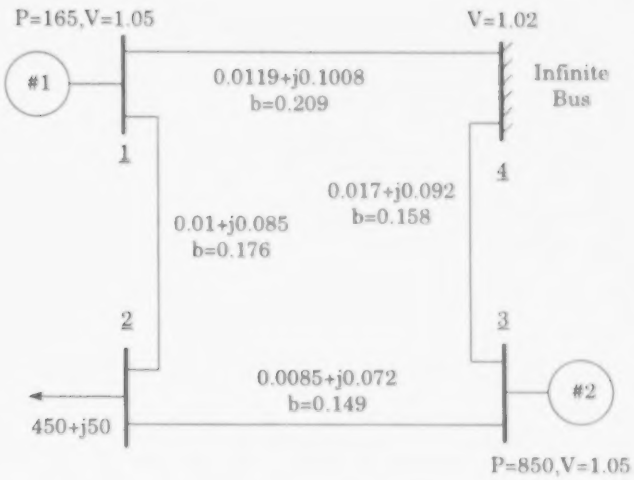


Fig. 8.1 Single–Line Diagram of the Two Machine Infinite Bus System

8.2.2 Stability Studies

(1) *Steady–State Stability*

In an interconnected power system, there are hundreds of oscillatory modes. Usually two distinct types of system oscillations, as has been outlined in the previous section, are recognized in an analysis of system stability and control problem. They are referred to as local plant modes and inter–area modes. The fundamental objective of excitation control is to provide maximum damping to both types of modes, and at the same time to enhance transient stability.

Part of this objective is accomplished by using supplementary excitation control through a conventional power system stabilizer(CPSS). For the purpose of designing a CPSS, an eigenanalysis was conducted for the study system. Two oscillatory modes were calculated. The local mode of the test

machine is $1.6928\,Hz$ and the inter-area mode is $0.7813\,Hz$. Based on the design procedure presented in [8-4], two CPSSs were designed and tuned, one for the test machine, and the other for the equivalent machine. They will be used in the performance studies later in this chapter. A study of the system's frequency response was also performed. The response was obtained by simulating the study system in TSSP in SIMULINK environment. The disturbance to the system is a three-phase to ground fault for 6-cycles applied to the terminal bus of generator #1. The speed change of the generator was measured, and sampled at a sampling rate of $66.67\,Hz$.

A 512-point fast Fourier transform(FFT) was applied to the sampled data sequence, and the power spectral density, a measurement of the energy at various frequencies, was calculated. Figure 8.2 shows the semi-logarithmic plot of the power spectral density. It can be seen that the two oscillatory modes are approximately $0.8\,Hz$ and $1.70\,Hz$. This is in agreement with what has been calculated from eigenanalysis.

(2) *Transient Stability*

In the studies reported in this section, a three-phase to ground fault for 6-cycles was applied on the line between buses #1 and #2 near the terminal bus of generator #1. The simulation model is given by Fig. 8.3, using TSSP under SIMULINK. In view of the large number of variables and study cases

involved, only the excitation voltage and rotor angle responses are shown in Figs. 8.4 and 8.5. The results of the transient simulations show that:

- For the case of CPSS with TGR, the inter–area mode oscillation is much pronounced (curve #4 of Fig. 8.5). The output of the exciter hits its upper limit only at the first swing(curve#4 of Fig. 8.4(b)). The overall performance of the excitation plus CPSS system can be improved by increasing the CPSS gain[8–24];
- For the case of CPSS without TGR, the excitation plus CPSS system improves damping and transient stability on both oscillatory modes(curve #3 of Fig. 8.5). The exciter reaches its ceiling voltages during the first two swings to provide maximum control action(curve #3 of Fig. 8.4(b));
- For the case of no CPSS, with TGR, the rotor angle oscillation is more damped and transient response is improved(curve #2 of Figs. 8.5 and 8.4(a)), as compared with the case of no TGR. This is true for local mode only. Its effect on the inter–area mode is imposed on to the local mode oscillation.
- For the case of no PSS and no TGR, the system is highly oscillatory. This scenario will be used to test whether a PSS can stabilize the system.

Table 8.1 Load Flow Results: Node Power(MW+jMVar)

Case	Bus #	V (pu)	θ (deg)	Generation	Load
A	1	1.050	6.4265	165+j69	—
	2	0.992	4.7698	—	450+j50
	3	1.050	21.2538	850+j111	—
	4	1.020	0.0	-519.9+j91.2	—
B	1	0.977	-8.9523	125+j50	
	2	0.910	-23.9549	—	750+j50
	3	1.050	-2.9906	450+j261.1	—
	4	1.050	0.0	211.3-j23.75	—

Table 8.2 Load Flow Results: Branch Power(MW+jMVar)

Case	Line $i-j$	S_{ij}	S_{ji}	Losses
A	1–2	43.29+j57.10	-42.72-j70.58	0.58-j13.48
	2–3	-407.28+j20.58	421.67+j85.75	14.39+j106.34
	3–4	428+j25.56	-399.86+j111.6	28.47+j137.15
	1–4	121.7+j12	-120-j20.38	1.66-j8.34
B	1–2	180+j71	-271-j11.4	8.89+j59.9
	2–3	-478-j38.6	502+j224	23.6+j185.7
	3–4	-52+j36.78	53-j49.7	0.74-j12.9
	1–4	-155-j21.24	158+25.9	3.02+4.71

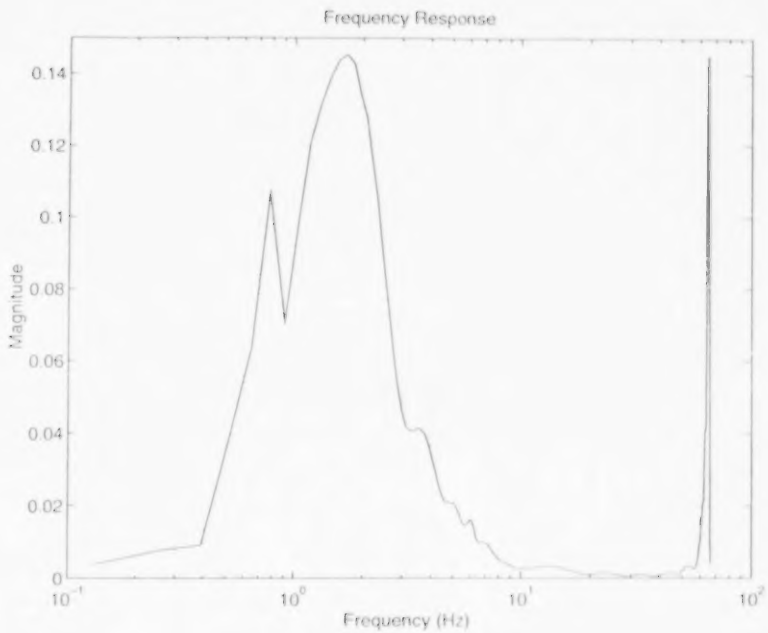


Fig. 8.2 Frequency Response Showing the Two Oscillatory Modes

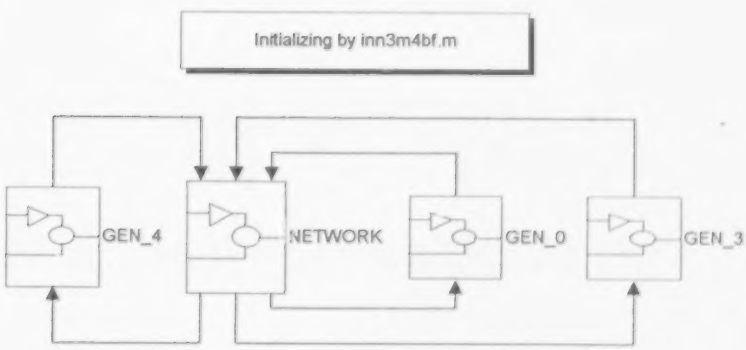


Fig. 8.3 Simulation Model of the Study System

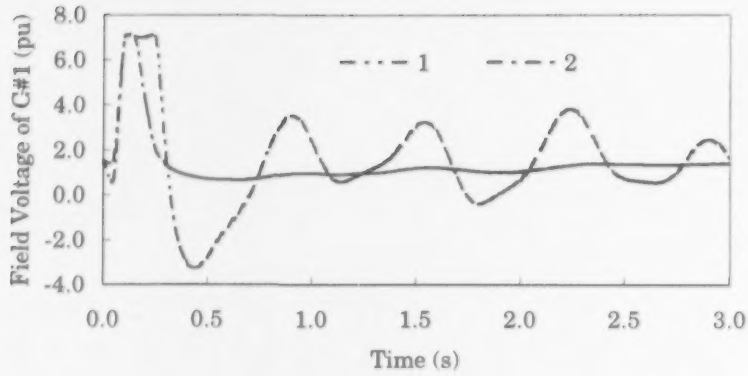


Fig. 8.4(a) Field Voltage Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1

Curve #1 —No Transient Gain on AVR, No CPSS

Curve #2 —With Transient Gain on AVR, No CPSS

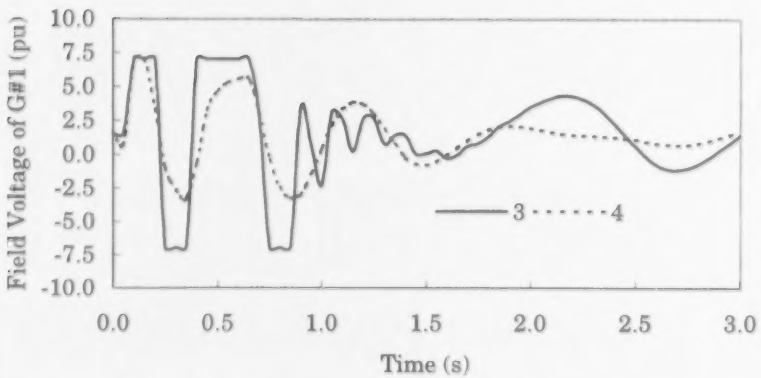


Fig. 8.4(b) Field Voltage Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1

Curve #3 —No Transient Gain on AVR, with CPSS

Curve #4 —With Transient Gain on AVR and CPSS

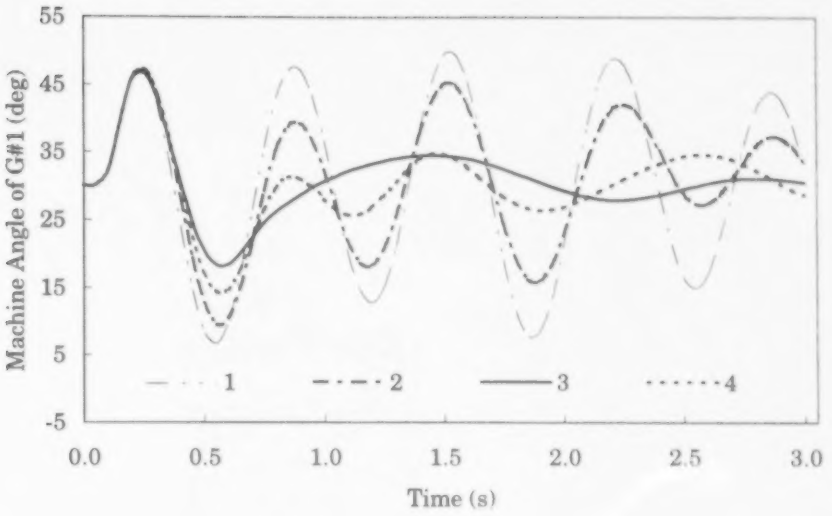


Fig. 8.5 Generator Angle Responses of the Test Machine, Following A 3–Phase To Ground Fault of 6–Cycles at the Terminal Bus of G#1 (Curve No. and description agree with those of Figs. 8.4(a) & (b))

8.3 DESIGN OF TWO ANN-FL PSSs

8.3.1 General Comments

The most difficult problem that a designer tries to solve is to make the best compromise in choosing PSS parameter values so that the controller will yield best performance when it faces environmental changes. Such a goal may never be truly realized in practice as the design is purely based on linear control theory where such a designed controller operates at its optimum only at the environment for which it is designed.

An hybrid of artificial neural network and fuzzy logic power system stabilizer(ANN–FL PSS) is a trained fuzzy logic controller. It combines the merits of both neural networks and fuzzy logic to form an intelligent control system. In this section, two ANN–FL power system stabilizers are designed, one using speed deviation of the generator as the input signal, the other using accelerating power.

The design procedure employed is similar to that used in Chapter 5, though, there are specific concerns in each design. The two controllers to be designed will have the same structure as that used in Chapter 5. The training program is similar to that of Chapter 5, with only minor modifications. The emphasis of this chapter will be focused on the performance of the two ANN–FL PSSs, when they are applied to a multimachine power system. Six aspects of information are needed to design an hybrid ANN–FL control scheme. They are outlined in the next section.

8.3.2 ANN–FL PSS Using Speed Deviation

(1) *Linguistic Variables*

As the objective of a power system stabilizer applied to a synchronous generator control is to damp rotor oscillations, it is natural to monitor the speed and to use it as a feedback control variable. Since speed deviation is a result of the unbalanced torque applied to the rotor shaft, the key to bringing the speed deviation to zero is to reduce the unbalanced torque to zero. This

residual torque equals the summation of the positive mechanical input torque from the prime mover and the negative electrical torque output from the generator. For the first few seconds after a system disturbance, due to technical reasons, the mechanical torque can not be changed dramatically to bring the residual torque to zero. It is the electrical torque that must be changed. The electrical torque is directly proportional to the excitation voltage and inversely proportional to the speed of the machine. Therefore, the excitation voltage could be chosen as the output variable of the controller to be designed. Only there is another problem; the excitation voltage is also controlled by a voltage regulator. The function of this control loop is to manage the reactive power flow in the system so that a desired voltage profile can be maintained during steady–state operations of the system.

To maintain the functionalities of the excitation system for both steady–state and transient operations, the control signal from the stabilizing controller can be introduced to the low signal front end of the excitation system as supplementary control. During steady–state operation, this supplementary signal must be zero. During transient operations, it must be able to control the excitation voltage to act according to the change of the speed. Another input variable to the controller can be the rate of speed change, as has been used in Chapter 5. Therefore, the two fuzzy inputs and one fuzzy output variables can be expressed as:

$$\Delta\omega = [NB, NM, NS, PS, PM, PB] \tag{8.1}$$

$$\Delta\omega / \Delta t = [NB, NM, NS, PS, PM, PB] \tag{8.2}$$

$$\Delta V = [N, P] \tag{8.3}$$

where the various linguistic labels are defined as negative big(NB), negative medium(NM), negative small(NS), positive small(PS), positive medium(PM), positive big(PB), negative(N), and positive(P).

(2) *Membership Functions*

By expert's knowledge, the ranges of change of these fuzzy variables defined above is known. Therefore, the universe of discourse of the fuzzy sets can be determined as follows:

$$\Delta\omega = [-0.1, +0.1](pu) \tag{8.4}$$

$$\Delta\omega / \Delta t = [-10, +10](pu / s) \tag{8.5}$$

$$\Delta V_s = [-0.4, +0.4](pu) \tag{8.6}$$

Six membership functions are assigned to each of the two input fuzzy variables, and two membership functions assigned to the output variable. The structure of the membership functions of the state variables $\Delta\omega / \Delta t$, $\Delta\omega$, and ΔV_s are the same as shown in Figs. 5.5 – 5.7, only the universes of discourse are different. Membership functions for fuzzy sets $A_{i2}, \ldots, A_{i5} (i = 1,2)$ are expressed by triangular functions which take the form of (8.7), shown in Fig. 8.6, and membership functions for fuzzy sets $A_{i1}, A_{i6}, B_i (i = 1,2)$ are

expressed by trapezoidal functions which take the form of (8.8), shown in Fig. 8.7. Table 5.1 lists eighteen rules used as the rule base in designing the ANN–FL power system stabilizers.

$$f(x,a,b,c)=\begin{cases}\dfrac{x-a}{c-a}, a\le x\le c,\\ \dfrac{b-x}{b-c}, c\le x\le b,\\ 0 \quad , otherwise.\end{cases} \tag{8.7}$$

$$f(x,a,b,c,d)=\begin{cases}\dfrac{x-a}{b-a}, a\le x\le b,\\ 1 \quad , b\le x\le c,\\ \dfrac{d-x}{d-c}, c\le x\le d,\\ 0 \quad , otherwise.\end{cases} \tag{8.8}$$

(3) *Data Collection and Training*

A number of simulation studies were carried out on the two–machine power system, using both the speed power system stabilizer(PSS) and the accelerating power PSS similar to those of [8–19]. The disturbances used in the studies were (a) a three–phase to ground fault at the terminal bus of generator #1 for 6 cycles; (b) a step reference voltage change of +4% to the automatic voltage regulator(AVR). The load level was changed to 100%, 75%, and 50%, respectively, when the three–phase fault was applied to the system. Each simulation lasted three seconds. The obtained time–domain responses were sampled with a sampling frequency of 100 Hz. The sampled results

were saved in text files as listed in Table 8.3 for compiling training and checking data sets.

(4) *Display of the Trained ANN–FL PSS*

A structure of five–layer feedforward network similar to that of Fig. 5.8 is used as the ANN–FL PSS controller. The universe of discourse of each input variable is initially divided into six segments. Each segment is assigned a membership function with initial parameters. This five–layer feedforward network was then trained using the collected data set and checked. Figures 8.8–8.10 show the trained membership functions of the controller.

8.3.3 ANN–FL PSS Using Accelerating Power

Though conventional PSSs using speed deviation have been successfully applied in practical systems, they can cause instability of torsional oscillatory modes[8–24]. One way to overcome this limitation is to use a torsional filter. It is a disadvantage to have to use such a filter which introduces a phase–lag at low frequencies and has a destabilizing effect on the exciter mode. These restrictions may or may not exist in applications of the newly–designed ANN–FL PSSs. Further investigation and practical implementation experience is needed to clarify this point.

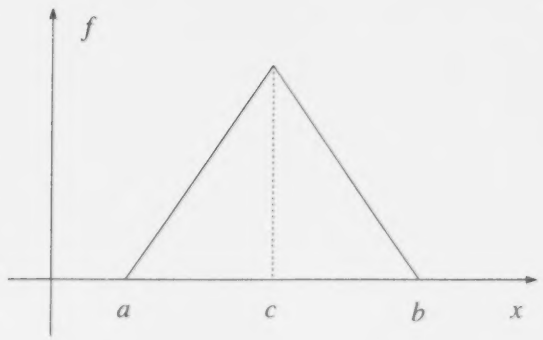


Fig. 8.6 Triangular Membership Function

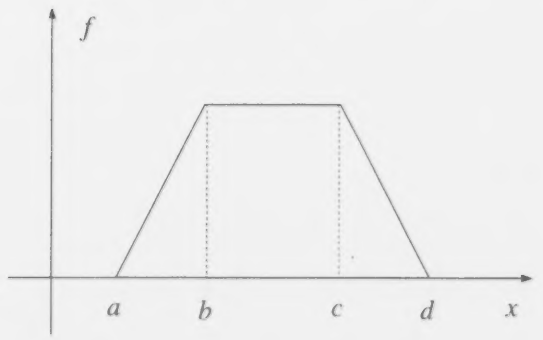


Fig. 8.7 Trapezoidal Membership Function

Table 8.3 Sources of Data Collection

Loading Level(%)	Conventional PSS	
	Speed	Acc. Power
100	scpss1	apcpss1
75	scpss7	apcpss7
50	scpss5	apcpss5
Step(+4%)	sstep1	apstep1

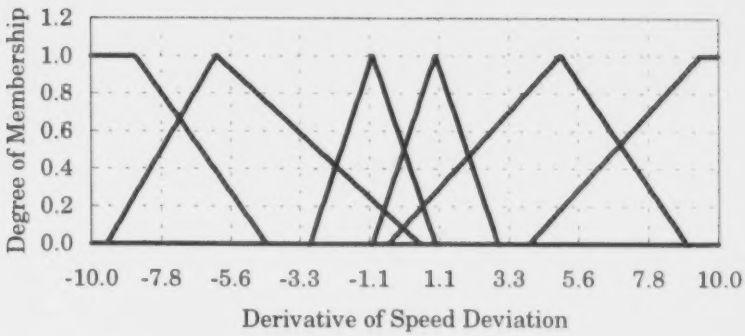


Fig. 8.8 Membership Functions for State Variable $\Delta\omega / \Delta t$

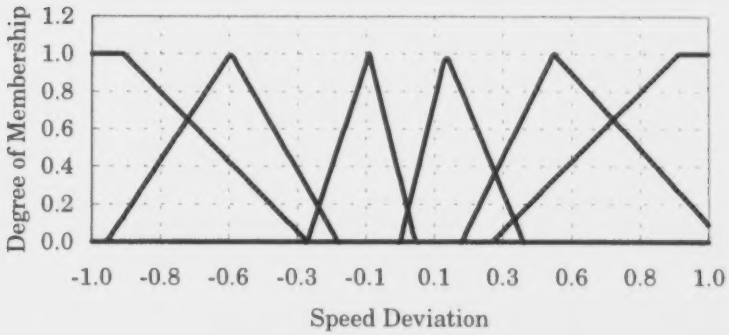


Fig. 8.9 Membership Functions for State Variable $\Delta\omega$

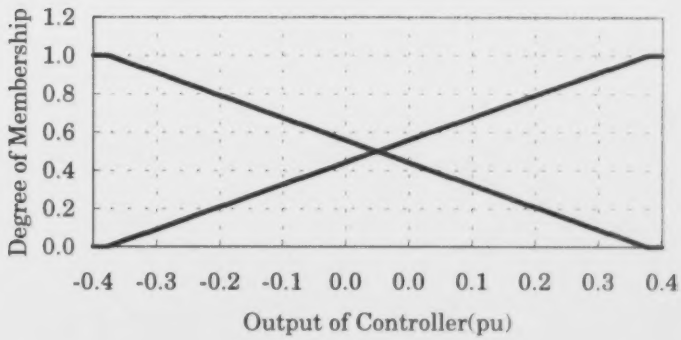


Fig. 8.10 Membership Functions for Output Variable ΔV

To overcome the limitations of the CPSS based on speed deviation, the Delta–P–omega stabilizers have been developed[8–25]. The principle of this stabilizer is illustrated by (8.9) where an equivalent signal to speed deviation has been derived from the accelerating power. This is given by,

$$\Delta\omega_{eq} = \frac{1}{2H}\int(\Delta P_m - \Delta P_e)dt \tag{8.9}$$

where

$2H$ = inertia constant of the machine,

ΔP_m = change in mechanical power,

ΔP_e = change in electrical power, and

$\Delta\omega_{eq}$ = equivalent speed deviation.

Since torsional components are inherently attenuated in the integral of ΔP_e signal, $\Delta\omega_{eq}$ will be free of torsional modes if the ΔP_m component is negligible as in many applications. Otherwise there is a technique to measure the integral of ΔP_m without introducing torsional modes[8–24].

On the basis of similar considerations for designing an accelerating power CPSS, an ANN–FL PSS using accelerating power as the input signal was designed in this research. The design procedure, controller structure and training program are basically the same as those used in designing the speed ANN–FL PSS. Therefore, only its simulation studies will be presented in the following section.

8.4 PERFORMANCE EVALUATION OF THE ANN-FL PSS

8.4.1 General Comments

In this section, a comprehensive investigation of the performance of the two newly–designed ANN–FL PSSs, when applied to the two–machine infinite–bus power system, is reported. Simulation studies were carried out for two different operating conditions, demonstrating the robustness of the PSSs. Specifically, the following categories of disturbances were used in the study: (1) a 4% step reference voltage change, simulating the effect of voltage variation in the system; (2) a 0.25pu step increase of the input power to generator #1, simulating the transients of load variation; and (3) severe three–phase to ground faults at different locations with different switching sequences, simulating transient performance of the various PSSs. Load flow Case A in Tables 8.1 & 8.2 was used unless otherwise stated in the presentation that follows.

8.4.2 Step Reference Voltage Change

To illustrate the dynamic behavior of the power system when it experiences a slight shift of the operating condition from one point to another, a step reference voltage of 4% increase was applied at the terminal bus of G#1. This disturbance causes both transient and steady–state changes in the rotor angle, generated reactive power and terminal voltage of the machine. For the

purpose of simulation studies, the transient gain reduction(TGR) of the excitation system of G#1 was disabled in order to make the oscillations more pronounced, so that the advantages of each control scheme can be more readily seen. Due to space limitation, only partial results obtained from the simulations are shown in Figs. 8.11(a) ~ 8.11(d). The following observations include also the results not shown in figures.

- When a generator is required to be operating from one reference voltage to another, say from 1.05 pu to 1.09 pu in this example, it is necessary to generate more reactive power into the system to maintain that desired voltage profile. Figures 8.11(b) & 8.11(c) show that the terminal voltage of G#1 is stabilized at 1.09 pu and the reactive power is raised from 0.7 pu to 1.4 pu in two seconds after the disturbance.

- It is shown in the simulations that the real power of G#1 returned to its original operating point in two seconds after the disturbance(not illustrated). But the power angle is reduced from 30° to 25°, as shown in Fig. 8.11(a).

- The control contribution of the CPSS and the ANN-FL PSS to the stabilization of the oscillations is basically the same; the latter offers a marginal advantage over the CPSS. This is seen from the output of the excitation system shown in Fig. 8.11(d) and from other responses.

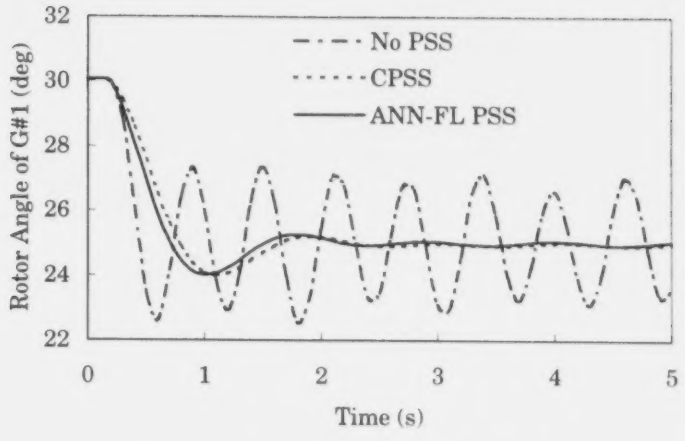


Fig. 8.11(a) Machine Angle Response to A 4% Step Voltage Change

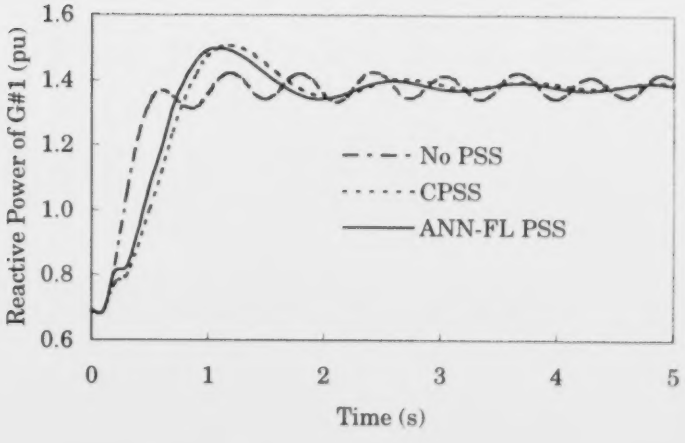


Fig. 8.11(b) Reactive Power Response to A 4% Step Voltage Change

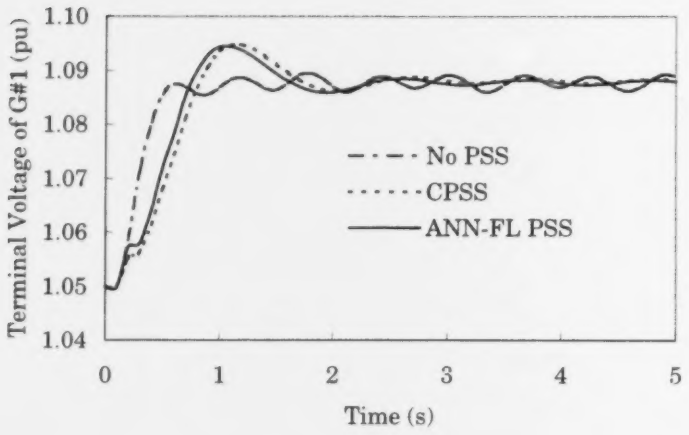


Fig. 8.11(c) Terminal Voltage Response to A 4% Step Voltage Change

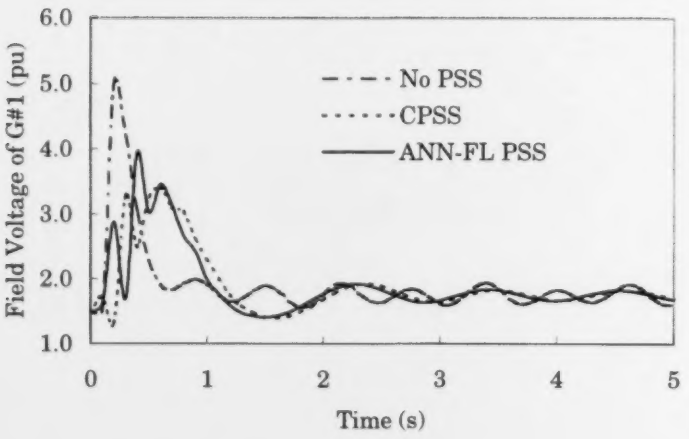


Fig. 8.11(d) Field Voltage Response to A 4% Step Voltage Change

8.4.3 Step Load Change

Load variation is present in every power system. It is intended that the supplementary excitation control can provide a means to accelerate the stabilization process after a sudden load change. To illustrate the performance of the CPSS and the newly–designed ANN–FL PSS under a load disturbance, a step increase of 0.25 pu of the input power was applied to generator #1. In terms of dynamic behavior of the generator, applying this disturbance is equivalent to losing a local load. It causes the machine angle and electric power to shift from the original operating points to new ones. Figures 8.12(a) ~ 8.12(d) show part of the results obtained from three simulations. The following observations are in order:

- Since the input mechanical power is increased by 0.25 pu, the output of the electric power from generator #1 is also increased by the same amount as shown in Fig. 8.12(b), from 1.65pu to 1.90pu. This is equivalent to saying, if the mechanical power is kept unchanged, a sudden loss of 0.25 pu of the local load means that this amount of electric power has to be exported to the system.
- The control action of the ANN–FL PSS is instantaneous as shown in Fig. 8.12(d). Its performance is marginally advantageous over that of the CPSS, as seen in Figs. 8.12(a) & 8.12(c), showing the rotor angle and the terminal voltage response, respectively.

- The local mode oscillation of the electric power was damped out in less than one second (Fig. 8.12(b)). The rotor angle settled down to its new operating point in about two seconds (Fig. 8.12(a)). The terminal voltage experienced a maximum of 1.5% excursion during the first cycle after the disturbance and returned to its original operating point in two seconds. Figure 8.12(c) also shows the inter–area oscillation which can not be seen in Figs. 8.12(a) & 8.12(b) because of the $y-$axis scale. The amplitude of the oscillation is also very small.

8.4.4 Fault Conditions

Power system stabilizers(PSSs) were originally designed to enhance small–signal stability of power systems. At the same time, they should perform satisfactorily under severe transient conditions. Transient stability is defined as the ability of a power system to maintain synchronism when subjected to a large disturbance such as a three–phase to ground fault, loss of generation, or loss of a large load. Under such a fault condition, large excursions of generator rotor angles, power flows and bus voltages can occur.

In the previous two sections, a step voltage change and loss of a fairly large load(0.25pu) were simulated. It has been shown that the newly–designed ANN–FL PSS demonstrated superior performance in the simulation studies. The objective of this section is to illustrate how the ANN–FL PSS performs under severe fault conditions.

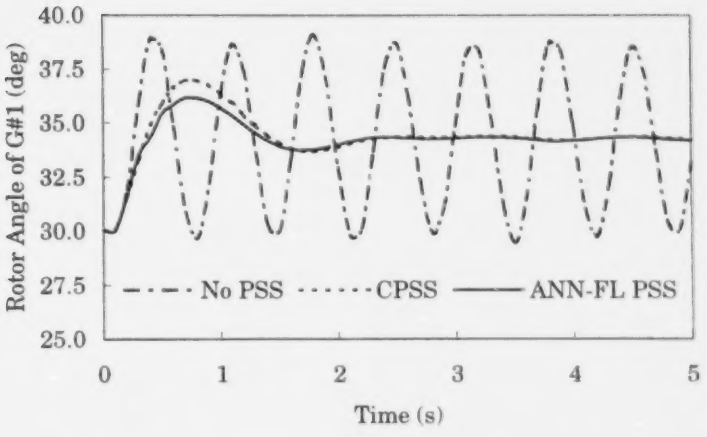


Fig. 8.12(a) Machine Angle Response to A 0.25pu Step Load Change

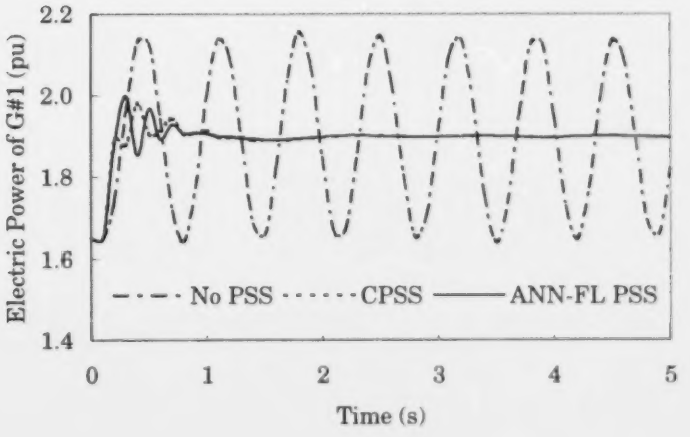


Fig. 8.12(b) Electric Power Response to A 0.25pu Step Load Change

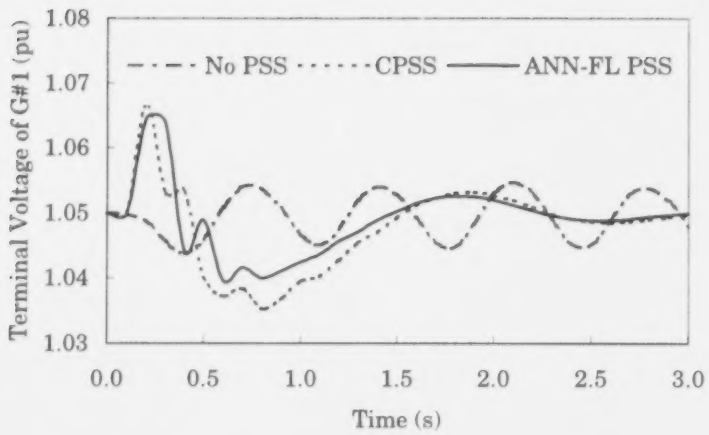


Fig. 8.12(c) Terminal Voltage Response to A 0.25pu Step Load Change

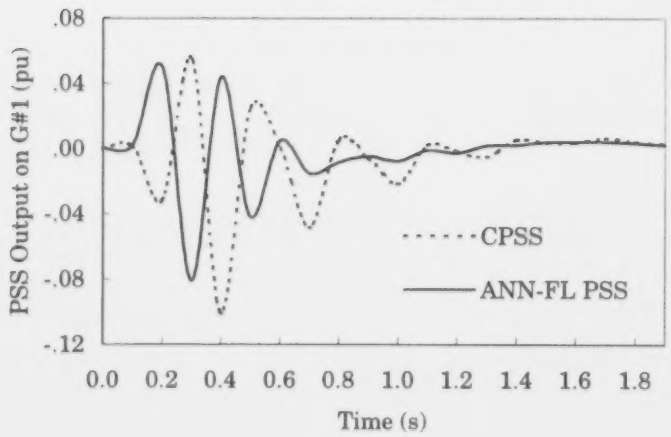


Fig. 8.12(d) Controller Response to A 0.25pu Step Load Change

(1) *Three–Phase Fault on Line 1–2*

With the same system configuration as used in the previous sections, six simulations were conducted of the system when a 6–cycle three–phase to ground fault on the line 1–2 near the terminal bus of generator #1 was applied. The line was restored right after the fault was cleared.

- Figures 8.13(a) & 8.13(b) show the situation where a PSS was activated on generator #1. It can be seen that the ANN–FL PSS provided more damping to the local mode than the CPSS did. The inter–area mode sustains(Fig. 8.13(a)).
- Figures 8.14(a) & 8.14(b) show the situation where each of the two generators was equipped with either the ANN–FL PSS or the CPSS. For the convenience of comparison, the response with only the ANN–FL PSS applied is also included(the fourth curve in each of the two figures). It can be seen that with the ANN–FL PSS on each machine or with the ANN–FL PSS on G#1 and the CPSS on G#2, both the local and the inter–area modes can be damped out in about two seconds.
- For this severe disturbance, one single PSS can not completely suppress the inter–area oscillation as shown in Fig. 8.13(a) and the #4 curve in Figs. 8.14(a) & 8.14(b).

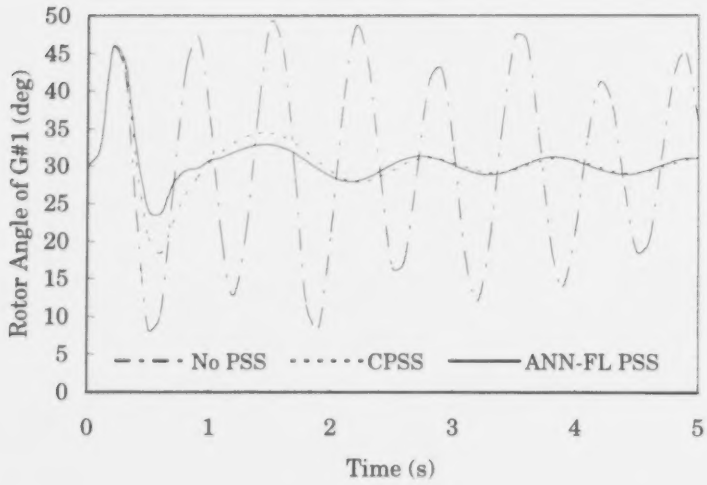


Fig. 8.13(a) Rotor Angle Response to A Three–Phase to Ground Fault at the Terminal Bus of G#1 for 6–Cycles, with One Controller

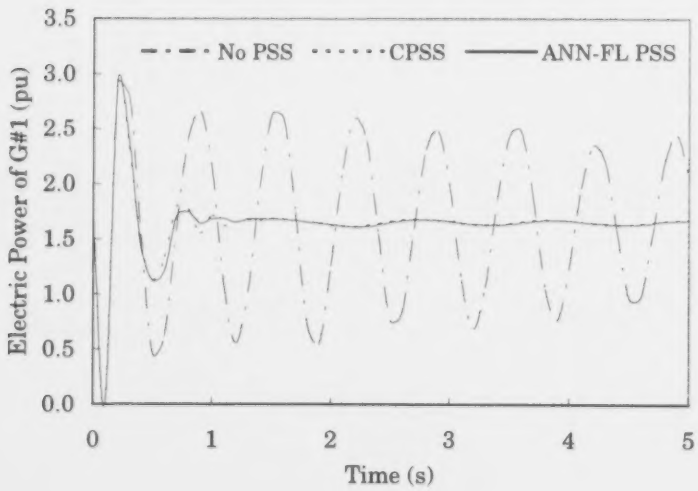


Fig. 8.13(b) Electric Power Response to A Three–Phase to Ground Fault at the Terminal Bus of G#1 for 6–Cycles, with One Controller

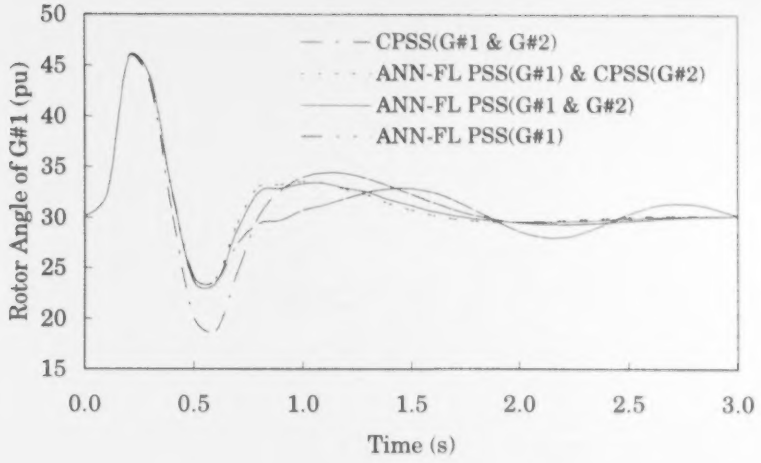


Fig. 8.14(a) Rotor Angle Response to A Three–Phase to Ground Fault at the Terminal Bus of G#1 for 6–Cycles, with Two Controllers

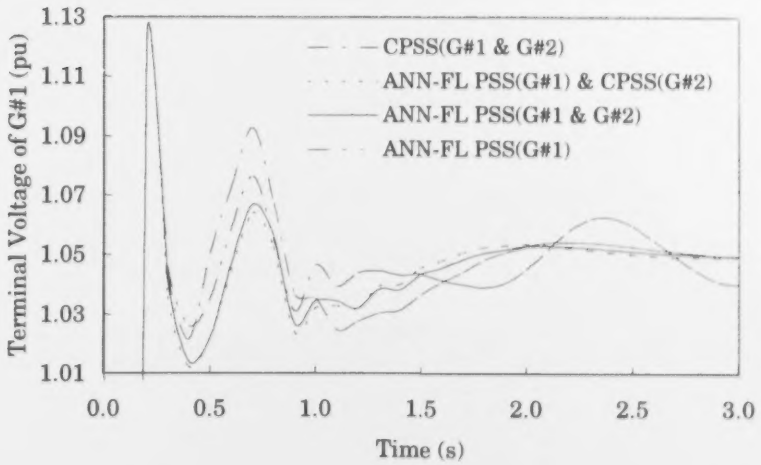


Fig. 8.14(b) Terminal Voltage Response to A Three–Phase to Ground Fault at the Terminal Bus of G#1 for 6–Cycles, with Two Controllers

(2) *Three–Phase Fault on Line 1–4*

The study system was then operated at a condition described by load flow Case B(Tables 8.1 & 8.2). A second three–phase to ground fault was simulated on the line 1–4 near bus #1 for 6–cycles. The line was restored 0.4s after the fault was cleared. Figure 8.15 shows the rotor angle response of G#1 following the disturbance. Three simulations were executed.

- When there is no PSS installed, the generators, and therefore the system, is oscillatorily unstable. When the CPSS is present on G#1, the "overshoot" of the first swing of G#1 is increased substantially as compared with the case of no PSS. With the ANN–FL PSS present on G#1, both the local and the inter–area modes are appreciably damped.

(3) *Three–Phase Fault on Line 2–3*

With the system operating at load flow Case B(Tables 8.1 & 8.2), a third transient condition was simulated with a 6–cycle three–phase to ground on the line 2–3 near bus #2. The line was restored 0.5s after the fault was cleared. Figure 8.16 shows responses of the rotor angle of G#1 following the disturbance.

- As the power transfer on this line was very large(see Case B in Table 8.1 & 8.2), the magnitude of oscillation is quite large. The CPSS provided

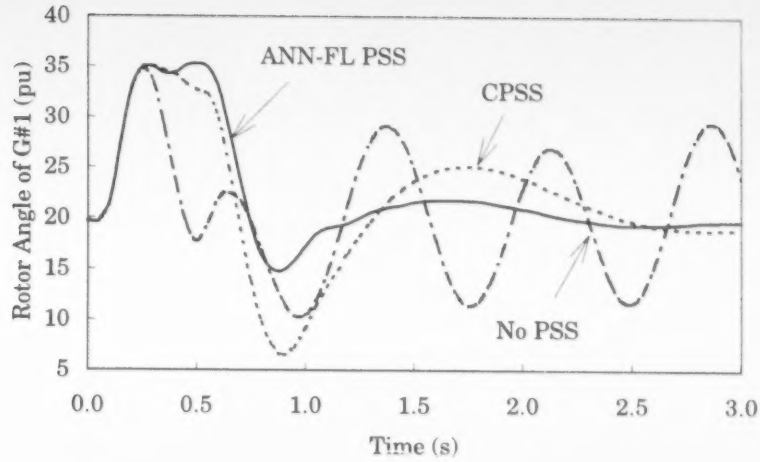


Fig. 8.15 Rotor Angle Response to A Three–Phase to Ground Fault on the Line 1–4 for 6–Cycles

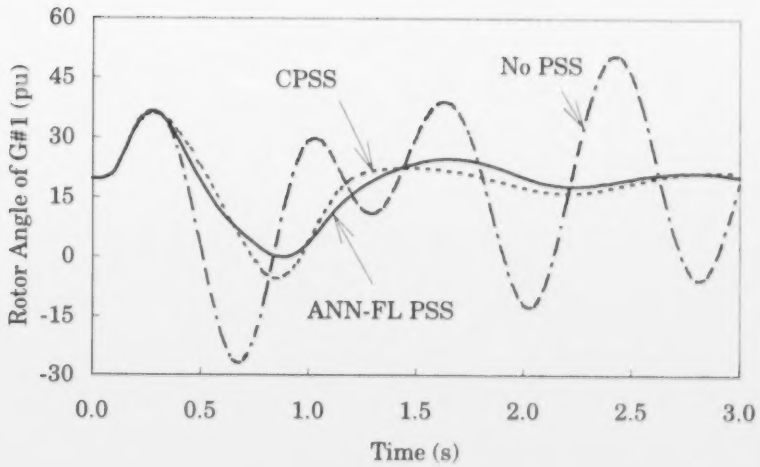


Fig. 8.16 Rotor Angle Response to A Three–Phase to Ground Fault on the Line 2–3 for 6–Cycles

substantial damping to the local mode. The ANN–FL PSS did an even better job(compare the three curves and note the y – axis scale too).

- The oscillation of the inter–area mode is manifest, due to the fact that the faulted line is directly connected to the equivalent machine.

8.4.5 ANN–FL PSS Using Accelerating Power

A transient simulation for the ANN–FL PSS using accelerating power as input signal was carried out. The fault applied was the 6–cycle three–phase to ground fault on line 1–2 at the terminal bus of G#1 as simulated previously. Figure 8.17 shows the rotor angle response of G#1 to the fault and the output of the PSS.

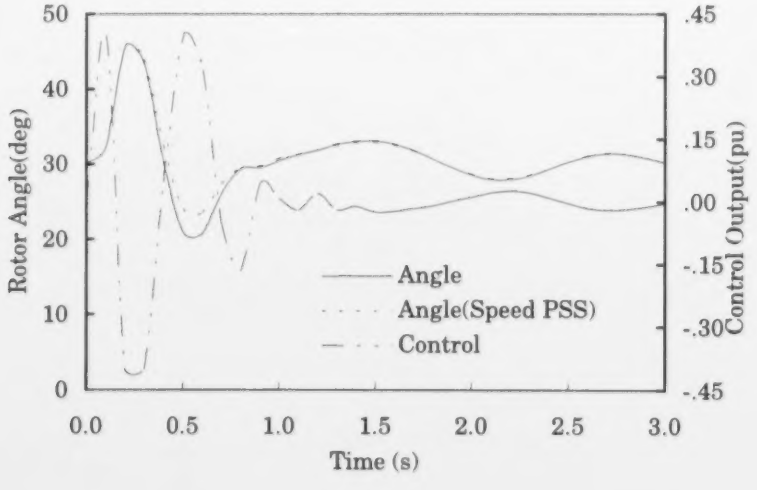


Fig. 8.17 Rotor Angle Response to the Three–Phase to Ground Fault on Line 1–2, with the ANN–FL PSS Using Accelerating Power As Input Signal

- It can be seen that the performance difference of the two ANN–FL PSSs is indistinguishable after the first swing in which the speed ANN–FL PSS produces less "overshoot".

8.5 SUMMARY

At the beginning of this chapter, an introduction to the enhancement of power system stability has been presented. At present, small– and large–signal stability problems are handled with the use of power system stabilizers (PSSs) which are cost–effective. A brief description of the study system employed in this thesis is given, including its load flow studies, frequency response, and transient simulations. Followed is the design of two ANN–FL PSSs, one using speed deviation as input and the other using accelerating power. The design procedure and the structure of the ANN–FL PSSs are the same as used in Chapter 5. In the last section, performance evaluation is presented of the newly–designed ANN–FL PSSs under different system disturbances.

CHAPTER 9

CONCLUSIONS

9.1 THE INITIATIVE

The stability problem of a power system can be summarized as being whether or not the synchronous generators in the system can be successfully controlled, i.e., maintained in synchronism following major upsets to the system. Classical and modern control techniques using mathematical system theory, which deals with the analysis and synthesis of dynamical systems, have served remarkably well. However these techniques are inadequate when one is confronted with the control of complex systems characterized by poor models, high dimensionality of the decision space, multiple performance criteria, time–varying parameter values, nonlinearities, disturbances, and failing component parts. As the system to be controlled becomes more complex, so does the complexity associated with the computation of the control law and the task of implementing the control in a timely fashion.

The research reported in this thesis was motivated by the fact that stability problems exist in any given power system and its control is a challenging task to engineers due to its increased complexity. Fixed parameter control systems that are designed using linear control theory may

not perform satisfactorily when the system experiences severe transients, since power systems are highly nonlinear.

This thesis has addressed a single issue in the category of power system stability control, that is, stability enhancement through supplementary excitation control accomplished with power system stabilizers(PSSs). The research on, and practice of, this technology has progressed from fixed parameter conventional PSSs, adaptive PSSs to intelligent control systems. The latter is at its early stage of application. An intelligent control system is one that can sense the environmental changes, process the sensor data, generate and execute a timely control action that guides the system from an initial state to a terminal state satisfying the various constraints and objectives imposed. Since intelligent control, defined as a combination of control theory, operations research and artificial intelligence, offers fast and robust control, it is emerging as one of the most popular new technologies in the industrial and manufacturing worlds. This has put an urgency to utility companies where new technology evolution is slow due to the fact that implementation of unproved new technology is a risk to both the customer and the utility. On the other hand, utilities want to improve their facilities and technology to better serve the customer.

9.2 REALIZATION

This thesis has presented a non–conventional control strategy consisting of an hybrid neural fuzzy–logic modeling technique, to meet the diverse demands of supplementary excitation control to enhance power system stability. Two ANN–FL PSSs have been designed for a two–machine infinite–bus power system and extensive simulation studies were performed using the Transient Stability Simulation Package(TSSP) implemented under SIMULINK. The following points summarize the observations from the work.

1. Steady–state stability studies have shown that the two–machine infinite–bus power system exhibits two oscillation modes. The local mode of the test machine is $1.7Hz$. The inter–area mode is $0.8Hz$. A study of the frequency response of the system was carried out to confirm the obtained results.
2. Transient simulation studies have shown that the multimachine power system is insufficiently damped and highly oscillatory without supplementary excitation control. This condition is designed for the purpose of testing any PSS to see whether it can stabilize the system.
3. When the test machine experiences a 4% step increase of its reference terminal voltage, the operating point of the machine has to change. The transient of this change has been simulated with different control schemes. It is shown that the system is oscillatory without PSS, and

stable with either the CPSS or the speed ANN–FL PSS. The latter provides marginally better performance.

4. When the test machine experiences a 0.25% step increase of its input mechanical power, both the CPSS and the speed ANN–FL PSS can stabilize the system. The latter provides instant control action(as it has no dynamic itself) and moderately more damping to the local mode.
5. When the system experiences severe disturbances such as three–phase to ground faults under different operating conditions, the ANN–FL PSSs exhibit much improved performance over the CPSS.
6. In Chapter 5, an ANN–FL controller is designed for speed–tracking control of a DC motor. It has been shown that the ANN–FL controller outperforms an optimally–tuned PID controller under various transients.

9.3 SIGNIFICANCE OF THE WORK

The results presented in this thesis have contributed to the innovation of design techniques for supplementary excitation control of synchronous generators. The hybrid neural fuzzy–logic modeling methodology combines the power of artificial neural networks(ANNs) and fuzzy–logic(FL) to form a control system that can learn, generalize and adapt. At the same time it can store knowledge of the system behavior as a structured database for later use. The significance of this research work can be summarized as follows:

1. This thesis has presented an innovative modeling technique referred to as ANN–FL modeling for designing supplementary excitation control systems to enhance the stability of multimachine power systems. Two ANN–FL PSSs, one using speed deviation as input, the other using accelerating power, are designed. Simulation studies have shown that this ANN–FL algorithm is fast in responding to state variable changes and immune to changes in the operating conditions.
2. An ANN–FL controller is designed for speed–tracking control of a DC motor. Its performance is compared with that of a PID controller. It is shown that the ANN–FL controller provides perfect speed–tracking control.
3. Simulation studies have shown that an ANN–FL controller designed for one machine can be utilized on another. This means that fast and economic intelligent controllers for both motor drives and generator control can be developed without customized design.
4. Handling of generator saliencies has been a challenge and an iterative procedure has been utilized in transient stability studies[9–1]. An innovative simulation algorithm overcoming this difficulty has been developed in this research.
5. A highly productive simulation environment has been developed as part of this research, based on the Transient Stability Simulation Package(TSSP) under the platform of SIMULINK. A device library has been created for

power system apparatuses and network equations. In this environment, advanced control systems can be easily designed, prototyped and simulated.

9.4 Future Work

Simulation studies have shown promising performance of the newly–designed ANN–FL controller for DC motor control and ANN–FL power system stabilizers for synchronous generator control. There are a few areas where further research work can be carried out.

1. Investigate whether there exist torsional oscillatory modes with an ANN–FL PSS using speed deviation as input. If such oscillatory modes exist, accelerating power may be utilized as input signal, since the integral of accelerating power is free of torsional modes[9–2].
2. Investigate whether coordinated operation[9–3] of ANN–FL PSSs in a multimachine power system is beneficial to the damping of local and inter–area modes where synchronous generators can be divided into coherent generating groups[9–4].
3. Implement in laboratory the ANN–FL controller for DC motor control and the ANN–FL PSSs for synchronous generator control.
4. Investigate the applicability of the proposed ANN–FL algorithm in (a) steam temperature control on the boiler side in a generation station and (b) governor and combined governor and AVR control.

REFERENCES

CHAPTER 1

[1–1] P. Kundur, *Power System Stability and Control*, McGraw-Hill, Inc., 1994.

[1–2] F. P. de Mello and C. Concordia, "Concepts of Synchronous machine Stability as Affected by Excitation Control," *Stability of Large Electric Power Systems*, Edited by R. T. Byerly, E. W. Kimbark, IEEE Press 1974.

[1–3] A. K. Laha and K. E. Bollinger, "Power Stabilizer Design Using Pole-Placement Techniques on Approximate Power System Models," *IEE Proc.*, 122(9), September 1975, pp. 903-907.

[1–4] K. E. Bollinger, A. K. Laha, R. Hamilton, and T. Harras, "Power Stabilizer Design Using Root Locus Methods," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol.94, no. 5, September 1975, pp. 1484-1488.

[1–5] K. E. Bollinger and R. Lalonde, "Tuning Synchronous Generator Voltage Regulators Using On-Line Generator Models," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol.96, no. 1, January 1977, pp. 32-37.

[1–6] K. E. Bollinger, R. Winsor and A. Campbell, "Frequency Response Methods for Tuning Stabilizers to Damp Out Tie-Line Power Oscillations: Theory and Field-Test Results," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 98, no. 5, September 1979, pp. 1509-1515.

[1–7] K. E. Bollinger and A. F. Mistr Jr., "PSS Tuning at the Virginia Electric and Power Co., Bath County Pumped Storage Plant," *IEEE Trans. On Power Systems*, vol. 4, no. 2, May 1989, pp. 566-574.

[1–8] F. P. de Mello, P. J. Nolan, T. F. Laskowski, and J. M. Undrill, "Coordinated Application of Stabilizers in Multi-machine Power Systems," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 99, no. 3, May/June 1980, pp. 892-901.

[1–9] S. Abe and A. Doi, "A New Power System Stabilizer Synthesis in Multi-Machine Power Systems," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 102, no. 12, December 1983, pp.3910-3918.

[1–10] P. Barret, Y. Colot, M. Herouard, J. P. Meyer, J. Michard and J. P. Monville, "Modeling and Tests at Fessenheim Power Station of A 1080 MVA Turbo–generator and of Its Excitation System," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 100, no. 2, February 1981, pp. 3993-4006.

[1–11] Choo-Min Lim and S. Elangovan, "A New Stabilizer Design Technique for Multi-Machine Power Systems," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 104, no. 9, September 1985, pp.2393-2400.

[1–12] Pei-Hwa Huang and Yuan-Yih Hsu, "Eigenstructure Assignment in A Longitudinal Power System via Excitation Control," *IEEE Trans. on Power Systems*, PWRS vol. 5, no. 1, February 1990, pp. 96-101.

[1–13] Y. N. Yu and Q. H. Li, "Pole-Placement Power System Stabilizers Design of An Unstable Nine-Machine System," *IEEE Trans. on Power Systems*, PWRS-5, no.2, May 1990, pp.353-358.

[1–14] H. A. M. Moussa and Y. N. Yu, "Optimal Power System Stabilization Through Excitation and/or Governor Control," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol.91, no. 3, May/June 1972, pp. 1166-1174.

[1–15] Y. N. Yu and H. A. M. Moussa, "Optimal Stabilization of A Multi-Machine System," *IEEE Trans. on Power Apparatus and Systems,* PAS-vol. 91, no. 3, May/June 1972, pp. 1174-1182.

[1–16] W. J. Wilson, V. M. Raina, and J. H. Anderson, "Nonlinear Output Feedback Excitation Controller Design Based on Nonlinear Optimal Control and Identification Methods," *IEEE/PES Summer Meeting*, pages A76 343-4/1-9, July 1976.

[1–17] V. H. Quintana, M. A. Zohdy, and J. H. Anderson, "On the Design of Output Feedback Excitation Controllers of Synchronous Machine," *IEEE Trans. on Power Apparatus and Systems*, PAS-95, May/June 1976, pp. 954-961.

[1–18] E. H. Okongwu, W. J. Wilson, and J. H. Anderson, "Microalternator Stabilization Using A Physically Realizable Optimal Feedback Controller," *IEEE Trans. on Power Apparatus and Systems*, PAS-101, no. 10, October 1982, pp. 3771-3779.

[1–19] E. V. Larsen and D. A. Swann, "Applying Power System Stabilizers, Parts I, II and III," *IEEE Trans. on Power Apparatus and Systems,* PAS-100, no. 6, June 1981, pp. 3017-3046.

[1–20] W. E. Norum, *Multimode Damping of Synchronous Generator Transients Using Self-Tuning Control*, Ph.D. Dissertation, U of Alberta, Edmonton, Alberta, 1991.

[1–21] J. Y. Fan, T. H. Ortmeyer, R. Mukundan, "Power System Stability Improvement with Multivariable Self-Tuning Control," *IEEE Trans. on Power Systems*, PWRS, vol. 5, no. 1, February 1990, pp. 227-234.

[1–22] W. Gu, *Multivariable Self-Tuning Control of Synchronous Generator Systems*, Ph.D. Thesis, U of A, Edmonton, Alberta, 1989.

[1–23] S. Cheng, Y. S. Chow, O. P. Malik, G. S. Hope, "An Adaptive Synchronous Machine Stabilizer," *IEEE Trans. on Power Systems*, PWRS, vol. 1, no. 1, August 1986, pp. 101-109.

[1–24] Shi-Jie Cheng, O. P. Malik, G. S. Hope, "Self-Tuning Stabilizer for a Multimachine Power System," *IEE Proc.*, 133 pt C, May 1986, pp. 176-185.

[1–25] A. Ghosh, G. Ledwich, O. P. Malik, G. S. Hope, "Power System Stability Based on Adaptive Control Techniques," *IEEE Trans. on Power Apparatus and Systems*, PAS-vol. 103, no. 8, August 1984, pp. 1983-1989.

[1–26] M. A. Sheirah and M. M. Abd-El-Fattah, "Improved Load-Frequency Self-Tuning Regulator," *Inter. J. Control*, vol. 39, no. 1, January 1984, pp. 145-158.

[1–27] E. Irving, J. P. Barret, C. Charcossey, J. P. Monville, "Improving Power Network Stability and Unit Stress with Adaptive Generator Control," *Automatica*, vol. 15, no. 1, January 1979, pp. 31-46.

[1–28] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. On Neural Networks*, vol. 1, no.1, March 1990, pp. 4-27.

[1–29] W. T. Miller, III, R. S. Sutton and P. J. Werbos, Eds., Neural Networks for Control, The MIT Press, Cambridge, MA, 1992.

[1–30] D. J. Sobajic, Ed., Neural Network Computing for the Electric Power Industry, *Proceedings of the 1992 INNS Summer Workshop*.

[1–31] L. A. Zadeh, "Fuzzy Sets," *Information Control*, vol. 12, 1965, pp.338-353.

[1–32] T. Kobayashi, A. Yokoyama and Y. Sekine, " Nonlinear Adaptive Control of Synchronous Generator Using Neural Network Based Regulator," *Proceedings of ISAP* '94, vol. 1, pp. 55-61.

[1–33] Y. Zhang, O. P. Malik, G. S. Hope and G. P. Chen, " Application of an Inverse Input/Output Mapped ANN as a power System Stabilizer," *IEEE Transactions on Energy Conversion*, vol. 9 no. 3 Sept 1994, pp. 433-441.

[1–34] V. Ramamurthi and D. V. Sudhakar, "An Adaptive Forced Action Excitation Controller for Large Synchronous Generators in a Power System," *Proceedings of the International Symposium on Salient-Pole Machines with Particular Reference to Large-Size Hydro-Electric Generators and Large Synchronous Motors*, Huazhong Science & Technology University, Hubei, China.

[1–35] F. Beaufays and B. Widrow, "Load-Frequency Control Using Neural Networks," *Neural Network Computing for the Electric Power Industry*, Proc. of the 1992 INNS Summer Workshop, 1993.

[1–36] D. T. Lee, D. J. Sobajic and Y. H., Pao, "Adaptive Power System Control with Neural Networks," IJCNN International Joint Conference on Neural Networks, vol. 2, 1992, pp. 834-843.

[1–37] Q.H. Wu, B. W. Hogg and G. W. Irwin, "On-Line Training of Neural Network Model and Controller for Turbogenerators," Proceedings of the First International Forum on Applications of Neural Networks to Power Systems (Cat. no.91TH0374-9), 1991, pp.161-165.

[1–38] S. Z. Ao and K. E. Bollinger, "An Adaptive Neuro-Controller for Speed Control of A Synchronous Generator," Canadian Conference on Electrical and Computer Engineering, May 26-29,1996, Calgary, Canada. CCECE'96, pp. 582-585.

[1–39] T. Hiyama, M. Kugimya and H. Satoh, "Advanced PID Type Fuzzy Logic Power System Stabilizer," 94 WM 127-EC.

[1–40] T. Hiyama, S. Oniki and H. Nagashima, "Evaluation of Advanced Fuzzy Logic PSS on Analog Network Simulator and Actual Installation on Hydro Generator," 95 SM 453-EC.

[1–41] A. R. Hasan and A. H. M. Sadrul Ula, "Design and Implementation of a Fuzzy Controller Based Automatic Voltage Regulator for A Synchronous Generator," 94 WM 025-7 EC.

[1–42] H. C. Chang, M. H. Wang, "Neural Network-Based Self-Organizing Fuzzy Controller for Transient Stability of Multimachine Power Systems," IEEE Trans. on Energy Conversion, vol. 10, no. 2, June 1995, pp. 339-347.

[1–43] Z. Ao, R. J. Fleming and T. S. Sidhu, "A Transient Stability Simulation Package (TSSP) for Teaching and Research Purposes," 94 SM 383-0 PWRS, 1994 Summer Meeting, San Francisco, *IEEE Trans. on Power Systems*, PWRS, vol. 10, no. 1, February 1995, pp. 11-17.

CHAPTER 2

[2–1] S. Haykin, Neural Networks, *A Comprehensive Foundation*, Maxwell Macmillan, Canada, 1994.

[2–2] Artificial Neural Network Approaches in Guidance and Control, Paris, AGARD, 1991, pp.1-4.

[2–3] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proc. of the IEEE*, vol. 78, no. 9, September 1989, pp. 1415-1442.

[2–4] D. Nguyen and B. Widrow, "The Truck Back-Upper: An Example of Self-Learning in Neural Networks," *Proc. Intl. Joint Conf. on Neural Networks*, vol. 2, Wash., DC, June 1989, pp. 357-363.

[2–5] D. G. Bounds, P. J. Lioyd, B. Mathew and G. Waddell, "A Multilayer Perceptron Network for the Diagnosis of Low Back Pain," *Proc. 2nd Intl. Conf. on Neural Networks*, vol. 2, San Diego, CA, July 1988, pp. 481-489.

[2–6] R. Brooks, "A Robot that Walks," *Neural Computation*, vol. 1, 1989, pp. 253-262.

[2–7] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, April 1987.

[2–8] J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing: Foundations of Research*, Cambridge, MA: M.I.T. Press, 1988.

[2–9] D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing*, MA: M.I.T. Press, 1986.

[2–10] H. Demuth and M. Beale, *Neural Network TOOLBOX*, The Mathworks, Inc., User's Guide, 1994.

CHAPTER 3

[3–1] L. A. Zadeh, "Fuzzy Sets," *Information Control*, vol. 12, 1965, pp.338-353.

[3–2] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I, Part II," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 2, 1990, pp. 405-435.

[3–3] E. Cox, *The Fuzzy Systems handbook*, A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems, AP Professional, Cambridge, MA 02139, 1994.

[3–4] L. A. Zadeh, King-Sun Fu, Kokichi Tanaka and M. Shimura, *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, Inc., 1975.

[3–5] E. H. Mamdani and B. R. Gaines, Eds., *Fuzzy Reasoning and Its Applications*, Academic Press,1981.

[3–6] P. M. Larsen, "Industrial Applications of Fuzzy Logic Control," *International Journal Man Machine Studies*, vol. 12, no. 1, 1980, pp. 3-10.

[3–7] Y. Tsukamoto, "An Approach to Fuzzy Reasoning Methods," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade and R. R. Yager, Eds., Amsterdam: North-Holland, 1979.

[3–8] T. Terano, K. Asai, and M. Sugeno, *Fuzzy Systems Theory and Its Applications,* Academic Press, Inc., 1992.

[3–9] A. Kaufmann and M. M. Gupta, *Fuzzy Mathematical Models in Engineering and Management Science*, Elsevier Science Publishers B. V., The Netherlands, 1988.

CHAPTER 4

[4–1] M. Sugeno, Ed., *Industrial Applications of Fuzzy Control,* Amsterdam, North-Holland, 1985.

[4–2] I. G. Umbers and P. J. King, "An Analysis of Human-Decision Making in Cement Kiln Control and the implementation for Automation," *International Journal of Man, Machine Studies,* 12(1), 1980, pp. 11-13.

[4–3] J. A. Bernard, "use of Rule-Based System for Process Control," *IEEE Control System Magazine,* 8(5), 1988, pp.3-13.

[4–4] T. J. Procyk and E. H. Mandani, "A Linguistic Self-Organizing Process Controller," *Automatica*, 15(1), 1979, pp. 15-30.

[4–5] M. Sugeno and M. Nishida, "Fuzzy Control of Model Car," *Fuzzy Sets System*, 16, 1985, pp. 103-113.

[4–6] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Maxwell Macmillan Canada, 1994.

[4–7] B. Kosko, "Adaptive Inference in Fuzzy Knowledge Networks," *Proceedings 1987 Int. Joint Conf. Neural Networks II*, 1987, pp. 261-268.

[4–8] C. T. Lin and C. S. G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," *IEEE Trans. on Computers* C-40(12), 1991, pp. 1320-1336.

[4–9] B. Kosko, *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1992.

[4–10] H. Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks Survey and Future directions," *Proc. Int. Conf. Fuzzy Logic Neural Networks*, (112UKA'90), Iizuka, Japan, July 20-24, 1990, pp. 13-26.

[4–11] J. S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. on System, Man, and Cybernetics*, vol. 23, May/June 1993.

[4–12] R. R. Yager and L. A. Zadeh, Eds., *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold, N. Y., 1994, pp. 233-249.

CHAPTER 5

[5–1] M. S. Sarma, *Electric Machines*, Steady State Theory and Dynamic Performance, West Publishing Company, 1994.

[5–2] D. Rumelhart and J. Meclelland, Eds., *Parallel Data Processing*, vol. 1, Chapter 8, The M.I.T. Press, Cambridge, MA 1986.

[5–3] J. S. Roger Jang and Ned Gully, *Fuzzy Logic Toolbox*, for use with MATLAB, The Mathworks, Inc. 1995.

[5–4] H. Naitoh and S. Tadakuma, "Microprocessor-Based Adjustable-Speed DC Motor Drives using Model Reference Adaptive Control," *IEEE Trans. on Industry Application*, vol. IA-23, no. 2, March/April, 1987, pp. 313-318.

[5–5] S. Weerasooriya and M. A. Sharkawi, "Identification and Control of a DC Motor Using Back-Propagation Neural Networks," *IEEE Trans. on Energy Conversion*, vol. 6, no. 4, 1991, pp. 663-669.

[5–6] R. R. Yager and L. A. Zadeh, Eds., *Fuzzy Sets, Neural Networks, and Soft Computing,* Van Nostrand Reinhold, New York, 1994.

CHAPTER 6

[6–1] R. H. Park, "Two-Reaction Theory of Synchronous Machines," *Trans. AIEE, Part I*: pp. 716-730, July 1929; Part II: pp. 352-355, June 1933.

[6–2] T. J. Hammons and D. J. Winning, "Comparisons of Synchronous-Machine Models in the Study of the Transient Behavior of Electrical Power Systems," *Proc. IEE*, vol. 118, no. 10, pp. 1443-1458, October 1971.

[6–3] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, The Iowa State University Press, Ames, Iowa, U.S.A., 1977.

[6–4] IEEE Committee Report, "Computer Representation of Excitation Systems," *IEEE Trans. on Power Apparatus and Systems*, vol. 87, June 1968, pp. 1460-1464.

[6–5] IEEE Committee Report, "Excitation System Models for Power System Stability Studies," *IEEE Transactions on Power Apparatus and System*, vol. PAS-100, pp. 494-507, February 1981.

[6–6] IEEE Standard 421.5, "IEEE Recommended Practice for Excitation Models for Power System Stability Studies," August 1992.

[6–7] Z. Ao, R. J. Fleming and T. S. Sidhu, "A Transient Stability Simulation Package (TSSP) for Teaching and Research Purposes," 94 SM 383-0 PWRS, 1994 Summer Meeting, San Francisco, *IEEE Trans. on Power Systems*, PWRS, vol. 10, no. 1, February 1995, pp. 11-17.

[6–8] Z. Ao, *A Comprehensive Stability Investigation of the Athabasca-Points North Power System*, M. Sc. thesis, U of Saskatchewan, September 1993.

[6–9] IEEE Excitation Subcommittee, "Digital Excitation Technology –A review of Features, Functions and Benefits," *IEEE Summer Meeting, July 1995.*

[6–10] P. Kundur, *Power System Stability and Control*, McGraw-Hill, Inc., 1994.

[6–11] IEEE Committee Report, "Dynamic Models for Steam and Hydro Turbines in Power System Studies," *IEEE Transactions on Power Apparatus and System*, vol. PAS-92, pp. 1904-1915, November 1973.

[6–12] E. V. Larson and D. A. Swann, "Applying Power System Stabilizers, Part I: General Concepts," *IEEE Trans. PAS*, vol. 100, pp. 3017-3024, 1981.

[6–13] IEEE Task Force on Load Representation for Dynamic Performance, "Load Representation for Dynamic Performance Analysis," 92 WM 126-3 PWRD, pp. 1-11, 1992.

[6–14] D. S. Brereton, D. G. Lewis and C. C. Young, "Representation of Induction-Motor Loads During Power-System Stability Studies," *AIEE Transaction*, pp. 451-460, August 1957.

[6–15] Nanjing Technology Institute, *Power System Analysis*, China Power Industrial Press, 1979.

CHAPTER 7

[7–1] D. W. Olive, "Digital Simulation of Synchronous Machine Transients," *IEEE Transactions on Power Apparatus and System*, vol. PAS-87, August 1968, pp. 1669-1674.

[7–2] H. W. Dommel, N. Sato, "Fast Transient Stability Solutions," *IEEE Transactions on Power Apparatus and System*, vol. PAS-91, July 1972, pp. 1643-1650.

[7–3] R. B. I. Johnson, M. J. Short, B. J. Cory, "Improved Simulation Techniques for Power System Dynamics," *IEEE Transactions on Power System*, vol. 3, no. 4, 1988, pp. 1691-1698.

[7–4] Manitoba HVDC Research Centre, *EMTDC–Users Manual*, Winnipeg, Canada, 1988.

[7–5] J. D. Clover, "A Personal Computer package for Power Engineering Education," *IEEE Transactions on Power System*, vol. 3, no. 4, pp. 1864-72, November 1988.

[7–6] R. Bonert, "Interactive Simulation of Dynamic Systems on A Personal Computer to Support Teaching," *IEEE Trans. on Power Systems*, vol. 4, no. 1, pp. 380-383, February 1989.

[7–7] D. C. Yu, S. T. Chen and R. F. Bischke, "A PC-Oriented Interactive Graphic Simulation Package for Power System Study," *IEEE Trans. on Power Systems*, vol. 4, no. 1, pp. 353-360, February 1989.

[7–8] J. D. Glover and L F. Dow, "Student Design Projects in Power Engineering," *IEEE Transactions on Power System*, vol. 5, no. 4, November 1990, pp. 1390-1399.

[7–9] Z. Ao, R. J. Fleming and T. S. Sidhu, "A Transient Stability Simulation Package (TSSP) for Teaching and Research Purposes," 94 SM 383-0 PWRS, 1994 Summer Meeting, San Francisco, *IEEE Trans. on Power Systems*, PWRS, vol. 10, no. 1, February 1995, pp. 11-17.

[7–10] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, The Iowa State University Press, Ames, Iowa, U.S.A., 1977.

[7–11] K. E. Bollinger and S. Z. Ao, "PSS Performance as Affected by Its Output Limiter," 95 SM 449-9 EC, *IEEE Trans. on Energy Conversion*, vol. 11, no. 1, March 1996, pp. 118-124.

[7–12] Z. Ao, T. S. Sidhu and R. J. Fleming, "Stability Investigation of A Longitudinal Power System and Its Stabilization by A Coordinated Application of Power System Stabilizers," 94 WM 128-9-EC, 1994 Winter Meeting, New York, *IEEE Trans. on Energy Conversion*, vol. 9, no. 3, September 1994, pp. 466-474.

[7–13] L. Fox and D. F. Mayers, *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, 1987.

[7–14] IEEE Committee Report, "Excitation System Models for Power System Stability Studies," *IEEE Transactions on Power Apparatus and System*, vol. PAS-100, February 1981, pp. 494-507.

[7–15] IEEE Committee Report, "Dynamic Models for Steam and Hydro Turbines in Power System Studies," *IEEE Transactions on Power Apparatus and System*, vol. PAS92, November 1973, pp. 1904-1915.

[7–16] IEEE Task Force on Load Representation for Dynamic Performance, "Load Representation for Dynamic Performance Analysis," 92 WM 126-3 PWRD, 1992, pp. 1-11.

[7–17] J. A. Martinez, "How to Adapt the EMTP for Classroom Instruction," *IEEE Trans. PWRS*, vol. 7, no. 1, February 1992, pp. 351-358.

[7–18] P. Kundur, *Power System Stability and Control*, McGraw-Hill, Inc., 1994.

[7–19] *DSP Design and Simulation Using the SIMULINK DSP Blockset*, The Mathworks, Inc., 1996.

CHAPTER 8

[8–1] Y. Y. Hsu, S. W. Shyue and C. C. Su, "Low frequency Oscillations in longitudinal Power Systems: Experience with Dynamic Stability of Taiwan Power System," *IEEE Trans. on Power Systems,* vol. PWRS-2,no. 1, February 1986, pp. 92-100.

[8–2] V. Arcidiacono, E. Forrai and F.Saccomanno, "Studies on Damping of Electromechanical Oscillations in Multimachine Systems with Longitudinal Structure," *IEEE Trans. PAS,* vol. 95, March/April, 1976, pp. 450-460.

[8–3] A. Al-Said, N. Abu-Sheikhab, T. Hussein , R. Marconato and Scarpellini, "Dynamic Behavior of Jordan Power System in Isolated Operation," *CIGRE* Paper 39-201, 1990.

[8–4] Z. Ao, T. S. Sidhu, R. J. Fleming, "Stability Investigation of a Longitudinal Power System and Its Stabilization by a Coordinated Application of Power System Stabilizers," 94 WM 128-9 EC, *IEEE Trans. on Energy Conversion*, vol. 9, no. 3, September 1994, pp. 466-474.

[8–5] R. L. Cresap and J. F. Hauer, "Emergence of a New Swing Mode in the Western Power System," *IEEE Trans. PAS,* vol. 100, 1981, pp. 2037-2043.

[8–6] M. Klein, G. J. Rogers and P. Kundur, "A Fundamental Study of Inter-Area Oscillations in Power Systems," *IEEE Trans. on Power System*, vol. PWRS-6, no. 3, August 1991, pp. 914-921.

[8–7] V. Larson and D. A. Swann, "Applying Power System Stabilizers, Part I: General Concepts," *IEEE Trans. PAS*, vol. 100, 1981, pp. 3017-3024.

[8–8] E. V. Larson and D. A. Swann, "Applying Power System Stabilizers, Part I: Performance Objectives and Tuning Concepts," *IEEE Trans. PAS*, vol. 100, 1981, pp. 3025-3033.

[8–9] E. V. Larson and D. A. Swann, "Applying Power System Stabilizers, Part M: Practical Considerations," *IEEE Trans. PAS*, vol. 100,1981, pp. 3033-3046.

[8–10] R. T. Byerly, R. J. Bennon and D. E. Sherman, "Eigenvalue Analysis of Synchronizing Power Flow Oscillations in Large Electric Power Systems," IEEE Trans. PAS, vol. 101, 1982, pp. 235-243.

[8–11] D. Y. Wong, G. J. Rogcrs, B. Porreta and P. Kundur, "Eigenvalue Analysis of Very Large Power Systems," *IEEE Trans. on Power System*, vol. 3, no. 2, May 1988, pp. 472-480.

[8–12] C. Corcordia and F. P. de Mello, "Concepts of Synchronous machine Stability as Affected b .y Excitation Control," *IEEE Trans. PAS*, vol. 88, April 1969, pp. 316-329.

[8–13] F. P. de Mello, P. J. Nollan, T. F. Laskowski, and J. M. Undrill, "Coordinated Application of Stabilizers in Multimachine Power Systems," *IEEE Trans. PAS*, vol. 99,1980, pp. 892-901.

[8–14] A. Doi, and S. Abe, "Coordinated Synthesis of Power System Stabilizers in Multimachine Power Systems," *IEEE Trans. PAS*, vol. 103,1994, pp. 1473-1479.

[8–15] D. R. Ostojic, "Stabilization of Multimodal Electromechanical Oscillations by Coordinated Application of Power System Stabilizers," *IEEE Trans. on Power Systems*, vol. 6, no. 4, November 1991, pp. 1439-1445.

[8–16] T. Hiyama, "Coherency-based Identification of Optimum Site for Stabilizer Applications," *IEE Proceedings*, vol. 130, Part C, 1983, pp. 71-74.

[8–17] R. W. de Mello, R. Podmore and K. N. Stanton, "Coherency-based Dynamic Equivalents for Transient Stability Studies," *Final Report on Electric Power Research Institute Project* RP90-4, phase III, December 1974.

[8–18] D. R. Ostojic, "Identification of Optimum Site for Power System Stabilizer Applications," *IEE Proceedings*, vol. 135, Part C, 1988, pp. 416-419.

[8–19] K. E. Bollinger and S. Z. Ao, "Power System Stabilizer Performance as Affected by its Output Limiter," 95 SM 449-EC, *IEEE Trans. on Energy Conversion*, vol. 11, no. 1, March 1996, pp. 118-124.

[8–20] Y. Y. Hsu and C. H. Cheng, "Design of Fuzzy Power System Stabilizers for Multimachine Power Systems," *IEE Proceedings*, vol. 137, Pt. C, no. 3, May 1990.

[8–21] M.A.M. Hassan, O. P. Malik and G. S. Hope, "A Fuzzy Logic Based Stabilizer for a Synchronous Machine," *IEEE Trans. on Energy Conversion*, vol. 6, no. 3, September 1991, pp. 407-413.

[8–22] T. Hiyama, T. Sameshima and C. M. Lim, "Fuzzy Logic Stabilizer with Digital Compensation for Stability Enhancement of Multimachine Power System," *Third Symposium on Expert Systems Application to Power systems*, April 15, 1991, Tokyo, Kobe, Japan.

[8–23] A. Ishigama, T. Imoto, S. Kawamoto and T. Taniguchi, "Fuzzy and Optimal Control for Stabilizing Power System," *Electrical Engineering in Japan*, vol. 111, no. 2, 1991.

[8–24] P. Kundur, M. Klein, G. J. Rogers and M. S. Zywno, "Application of Power System Stabilizers for Enhancement of Overall System Stability," *IEEE Trans. on Power Systems*, vol. 4, no. 2, May1989, pp. 614-626.

[8–25] D. C. Lee and P. Kundur, "Advanced Excitation Control for Power System Stability Enhancement," *CIGRE* 38-01, 1986.

CHAPTER 9

[9–1] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions," *IEEE Transactions on Power Apparatus and System*, vol. PAS-91, July 1972, pp. 1643-1650.

[9–2] D. C. Lee and P. Kundur, "Advanced Excitation Control for Power System Stability Enhancement," *CIGRE* 38-01, 1986.

[9–3] Z. Ao, T. S. Sidhu and R. J. Fleming, "Stability Investigation of A Longitudinal Power System and Its Stabilization by A Coordinated Application of Power System Stabilizers," *IEEE Trans. on Energy Conversion*, vol. 9, no. 3, September 1994, pp. 466-474.

[9–4] T. Hiyama, "Coherency-Based Identification of Optimum Site for Stabilizer Applications," IEE Proceedings, vol. 130, Part C, 1983, pp. 71-74.

APPENDIX A SATURATED REACTANCE AND TIME CONSTANTS

A.1 Saturated Reactance

It is assumed that leakage flux does not contribute in any way to the saturation of the machine, and that the leakage flux itself does not saturate either. Thus the reactances on the two axes can be given by

$$x_d = x_\sigma + x_{ad} \tag{A.1}$$

$$x_q = x_\sigma + x_{aq} \tag{A.2}$$

The saturated values of the reactances in the above equations are calculated as follows:

$$x_d^{(0)} = x_\sigma + x_{ad}^{(0)} \tag{A.3}$$

$$x_q^{(0)} = x_\sigma + x_{aq}^{(0)} \tag{A.4}$$

$$x_{ad} = kx_{ad}^{(0)} \tag{A.5}$$

$$x_{aq} = kx_{aq}^{(0)} \tag{A.6}$$

$$x_d = x_\sigma + kx_{ad}^{(0)} = kx_d^{(0)} + (1-k)x_\sigma \tag{A.7}$$

$$x_q = x_\sigma + kx_{aq}^{(0)} = kx_q^{(0)} + (1-k)x_\sigma \tag{A.8}$$

$$x_{ad} = x_d - x_\sigma = k(x_d^{(0)} - x_\sigma) \tag{A.9}$$

where the subscript σ denotes flux leakage.

A.2 Time Constants

The single line diagrams of the excitation winding and the two damping windings on the rotor are used to derive the time constants. Figure A.1 represents the open circuit of the excitation winding, from which the transient time constant is given by

$$\tau'_{do} = \tau_f = \frac{\left(x_{f\sigma} + x_{ad}\right)}{r_f} \tag{A.10}$$

Then the saturated time constant can be found from

$$\frac{\tau'_{do}}{\tau'^{(0)}_{do}} = \frac{\dfrac{\left(x_{f\sigma} + x_{ad}\right)}{r_f}}{\dfrac{\left(x_{f\sigma} + x^{(0)}_{ad}\right)}{r_f}}$$

$$= \frac{x_{f\sigma} + k\left(x^{(0)}_d - x_\sigma\right)}{x_{f\sigma} + \left(x^{(0)}_d - x_\sigma\right)} \qquad \text{(A.11)}$$

$$= k + (1-k)\frac{\left(x'_d - x_\sigma\right)}{\left(x^{(0)}_d - x_\sigma\right)}$$

where (A.3) and (A.12)-(A.13) are used:

$$x_{ff} = x_{f\sigma} + x_{ad} \qquad \text{(A.12)}$$

$$x'_d = x_d - \frac{x^2_{ad}}{x_{ff}} \qquad \text{(A.13)}$$

It is assumed that $x_{f\sigma}$, the leakage flux of the excitation winding, does not saturate either.

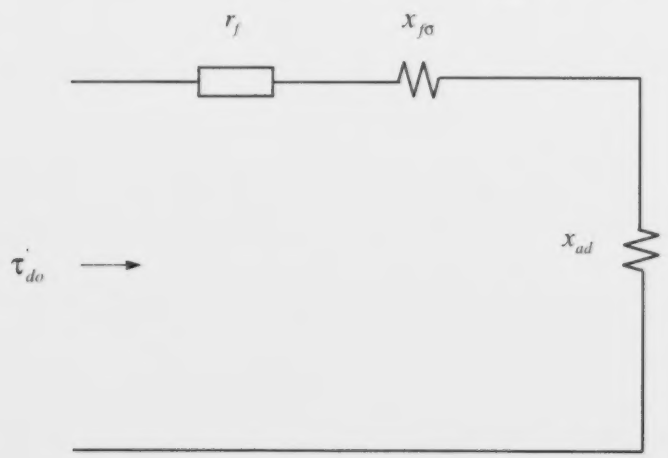


Figure A.1: Diagram for calculation of τ'_{do}

Figure A.2 represents the open circuit of the damper winding on $d-$axis, from which its subtransient time constant is calculated by

$$\tau''_{do} = \frac{\left(x_{D\sigma} + x_{ad} \| x_{f\sigma}\right)}{r_D} \tag{A.14}$$

Then its saturated value can be calculated by

$$\begin{aligned} \frac{\tau''_{do}}{\tau''^{(0)}_{do}} &= \frac{x_{D\sigma} + \left(x_{ad} \| x_{f\sigma}\right)}{x_{D\sigma} + \left(x^{(0)}_{ad} \| x_{f\sigma}\right)} \\ &= k + (1-k)\frac{\left(x''_d - x_\sigma\right)}{\left(x^{(0)}_d - x_\sigma\right)} \end{aligned} \tag{A.15}$$

where ' $\|$ ' denotes parallel connection of reactances. The following (A.16) is used in derivation of (A.15):

$$\begin{aligned} x''_d &= x_\sigma + \frac{x_{D\sigma} x_{f\sigma} x^{(0)}_{ad}}{x_{D\sigma}\left(x^{(0)}_{ad} + x_{\sigma\sigma}\right) + x_{ad} x_{f\sigma}} \\ &= x_\sigma + x_{D\sigma} \| x_{f\sigma} \| x^{(0)}_{ad} \end{aligned} \tag{A.16}$$

which is the equivalent reactance of Figure A.2 when the damper winding is short circuited and r_D is set to be zero.

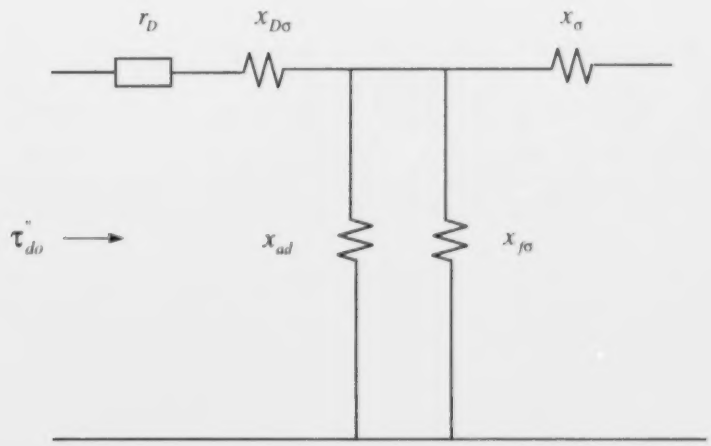


Figure A.2: Diagram for calculation of τ''_{do}

Figure A.3 represents the open circuit of the damper winding on $q-$axis, from which its subtransient time constant is calculated by

$$\tau_{q\sigma}^{''} = \frac{\left(x_{Q\sigma} + x_{aq}\right)}{r_Q} \tag{A.17}$$

$$\frac{\tau_{q\sigma}^{''}}{\tau_{q\sigma}^{''(0)}} = k + (1-k)\frac{\left(x_q^{''} - x_\sigma\right)}{\left(x_q^{(0)} - x_\sigma\right)} \tag{A.18}$$

The following (A.19) is used in derivation of (A.18):

$$x_q^{''} = x_\sigma + x_{Q\sigma} \| x_{aq}^{(0)} \tag{A.19}$$

which is the equivalent reactance of Figure A.3 when the damper winding is short circuited and r_Q is set to be zero.

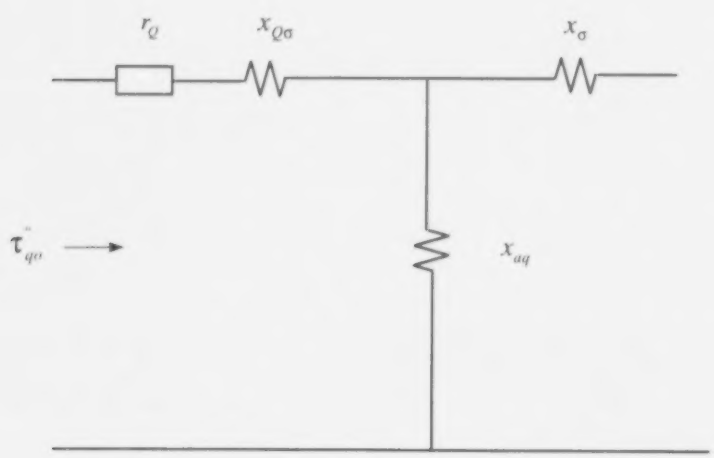


Figure A.3: Diagram for calculation of $\tau_{q\sigma}^{''}$

APPENDIX B M-FILE FIND_VIF.M

```
% Filename: find_vif.m,  network and generator interface
% Line fault can be introduced if desired.

% Inputs: state variables of devices
% Outputs: terminal voltages and currents

% Stephan Z. Ao
% CopyRight Feb 16, 1996

function vi_out = find_vif(u)

cmm
global Y  Yf Yp Yr BUSTYPE
global Ts Tp Tr Ifault
global Xds Xqs  Ts Tp Tr Y0

NBUS = max(size(Y));
A = zeros(2*NBUS,2*NBUS);                  %coefficient matrices
B = zeros(2*NBUS,1);
Zi = zeros(2,2);                           %impetance of generator i
Del = [];                                  %generator angle
Y0 =zeros(2*NBUS,2*NBUS);
nu = max(size(u));
TIME = u(nu);                              %last element of u is time
if (TIME<Ts)                               %prefault Y
  Y0 = Y;
elseif (TIME>=Ts & TIME<Tp )               %faulted Y
  Y0 = Yf;
elseif (TIME>=Tp & TIME<Tr )               %post-faulted Y
  Y0 = Yp;
elseif (TIME>=Tr )                         %restored Y
  Y0 = Yr;
end                                        %end if

for I=1:NBUS                               %decompose Y into real
for J=1:NBUS                               %and imaginary part
  A(2*I-1,2*J-1) = real(Y0(I,J));
  A(2*I-1,2*J  ) =-imag(Y0(I,J));
  A(2*I  ,2*J-1) = imag(Y0(I,J));
  A(2*I  ,2*J  ) = real(Y0(I,J));
end
end
uix = 1;                                   %point to the first element
I=1;                                       %of input u[]
for J=1:NBUS
id = BUSTYPE(J,2);
  if (id<0)
                                           %handle load buses here
  end
  if (id>=0)                               %handle generator buses
```

```
if (id==4)                                  %GEN_4
  Xdbar = XD11(I);
  Xqbar = XQ11(I);
  Epbar = [u(uix+1);                        %Eq"
           u(uix+2)];                       %Ed"
  ang = u(uix+3);                           %angle
  Del = [Del;ang];
  SX(I,2) = u(uix+3);
  SX(I,3) = u(uix);
  SX(I,4) = u(uix+2);
  SX(I,13) = u(uix+1);
  uix = uix + 4;

elseif (id==3)                              %GEN_3
  Xdbar = XD1(I);
  Xqbar = XQ1(I);
  Epbar = [u(uix);                          %Eq'
           u(uix+1)];                       %Ed'
  ang = u(uix+2);                           %angle
  Del = [Del;ang];
  SX(I,2) = u(uix+2);
  SX(I,3) = u(uix);
  SX(I,4) = u(uix+1);
  uix = uix + 3;

elseif (id==2)                              %GEN_2
  Xdbar = XD1(I);
  Xqbar = Xqs(I);
  Epbar = [u(uix); 0];                      %Eq', Ed'=0
  ang = u(uix+1);                           %angle
  Del = [Del;ang];
  SX(I,2) = u(uix+1);
  SX(I,3) = u(uix);
  SX(I,4) = 0.0;
  uix = uix + 2;

  elseif (id==1)                            %GEN_1
  Xdbar = Xds(I);
  Xqbar = Xqs(I);
  Epbar = [u(uix); 0];                      %Eq=cont, Ed=0
  ang = u(uix+1);                           %angle
  Del = [Del;ang];
  SX(I,2) = u(uix+1);
  SX(I,3) = u(uix);
  SX(I,4) = 0.0;
 uix = uix + 2;

 elseif (id==0)                             %Infinite Bus
  Xdbar = XD(I);
  Xqbar = XQ(I);
  Epbar = [u(uix); 0];                      %Vinf
  ang = u(uix+1);                           %angle
  Del = [Del;ang];
  SX(I,2) = u(uix+1);
```

```
    SX(I,3) = u(uix);
    SX(I,4) = 0.0;
   uix = uix + 2;
  end                                          %end internal if statement
Zi = [-R(I), -Xdbar;
       Xqbar, -R(I)];
T = [cos(ang)  sin(ang);                       %transformation
     sin(ang) -cos(ang)];
YIP = T*inv(Zi)*T;                             %impetance matrix
                                               %of generator i
                                               %modify A matrix

A(2*J-1,2*J-1) =A(2*J-1,2*J-1)-YIP(1,1);
 A(2*J-1,2*J  ) =A(2*J-1,2*J  )-YIP(1,2);
 A(2*J  ,2*J-1) =A(2*J  ,2*J-1)-YIP(2,1);
 A(2*J  ,2*J  ) =A(2*J  ,2*J  )-YIP(2,2);
vn = [u(nu-2);u(nu-1)];                        %voltage at slack bus
Zi = YIP*T*Epbar;                              %Yi'*Ti*E'--Zi as temp use
                                               % T as temp use
T  = [A(2*J-1,2*NBUS-1), A(2*J-1,2*NBUS);
      A(2*J  ,2*NBUS-1), A(2*J  ,2*NBUS)];

 TEM = T*vn;                                   %element of last column
                                               %of Y multiplied by vn

B(2*J-1,1) = -Zi(1,1) - TEM(1,1);
 B(2*J  ,1) = -Zi(2,1) - TEM(2,1);
 I = I+1;
 end                                           %end external if statement
 end                                           %end for  I loop
                                               %solve Ax = B equation
                                               %for voltage:Vxy--real&imag
                                               %Vcp--complex x+j*y vectors

 if (TIME>=Ts & TIME<Tp)                       %fault handling
   Iff = Ifault;
   AP = zeros(2*NBUS-4,2*NBUS-4);
        I1 = 1;
   for I=1:2*NBUS-2
    if ( (I<2*Iff-1) | (I>2*Iff) )

        J1 = 1;
     for J=1:2*NBUS-2
         if ( (J<2*Iff-1) | (J>2*Iff) )
         AP(I1,J1) = A(I,J);
         J1 = J1 + 1;
         end
     end                                       %for J
    BP(I1,1) = B(I,1);
    I1 = I1 + 1;
    end
   end

  VxyP = inv(AP(1:2*NBUS-4,1:2*NBUS-4))*BP(1:2*NBUS-4,1);

I1 = 1;
   for I=1:2*NBUS-2
```

```
   if (I<2*Iff-1 | I>2*Iff)
     Vxy(I,1) = VxyP(I1,1);
     I1 = I1 + 1;
   else
     Vxy(I,1) = 0.0;
   end;
   end                                          %for I
 end                                            %for fault period

 if (TIME<Ts | TIME>=Tp)
   Vxy = inv(A(1:2*NBUS-2,1:2*NBUS-2))*B(1:2*NBUS-2,1);
 end

Vcp = [];
   for I=1:NBUS-1
   Vcp = [Vcp;Vxy(2*I-1)+i*Vxy(2*I)];
   end
 Vcp = [Vcp;u(nu-2)];
 Vxy = [Vxy;u(nu-2);0];

 Ixy = Y*Vcp;                                   %calculate injected current
                                                %transform Ixy into
 vi_out = [];                                   %machine refence:Iqd=(2*NGEN-1,1)
I=1;
 for J=1:NBUS
   if (BUSTYPE(J,2)>=0)

   T = [cos(Del(I))  sin(Del(I));
        sin(Del(I)) -cos(Del(I))];
   Iqd = T*[real(Ixy(J));
            imag(Ixy(J))];

 vi_out = [vi_out;                              % output Iqd,Vxy &
     Iqd(1,1);Iqd(2,1);                         %Ixy for each generator
     Vxy(2*J-1,1);Vxy(2*J,1);
     real(Ixy(J));imag(Ixy(J))];
   I = I + 1;
   end
 end

%update saturated parameters

saturat

%END OF PROGRAM
```

APPENDIX C SPECIFICATIONS OF THE TWO–MACHINE INFINITE BUS SYSTEM

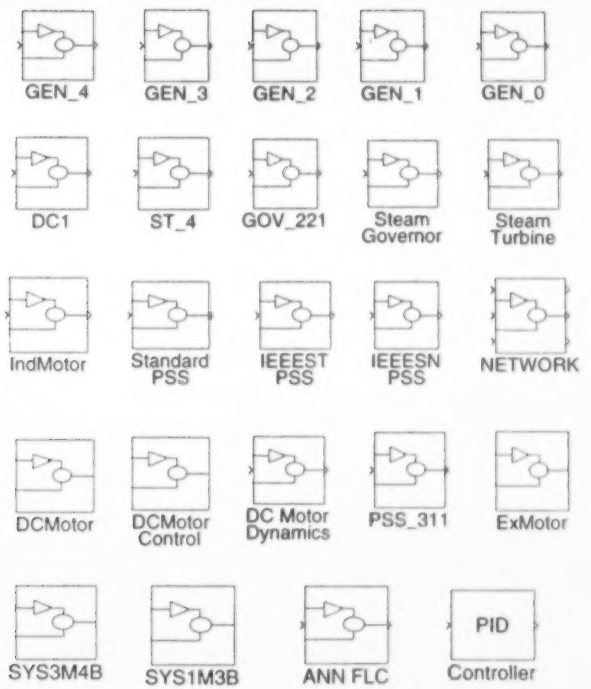
Table C.1 Parameters of the Two Generators

	Test Machine	Equivalent Machine
S_b	175.0	900.0
H	3.91	6.175
R_a	0.0032	0.0025
x_l	0.22	0.20
x_d	0.74	1.80
x_d'	0.51	0.30
x_d''	0.40	—
x_q	0.63	1.70
x_q'	—	0.55
x_q''	0.41	—
τ_{d0}'	4.73	8.00
τ_{d0}''	0.03	—
τ_{q0}'	—	0.40
τ_{q0}''	0.05	—
D	2.00	4.00
A_G	0.00003	0.00011
B_G	6.00	7.2

Table C.2 Parameters of the Excitation Systems

	Test Machine	Equivalent Machine
T_a	1.00	1.00
T_b	1.00–10.00	1.00–10.00
K_e	115.0	60.0
T_e	0.05	0.05
$E_{fd\,\min}$	–7.00	–7.00
$E_{fd\,\max}$	7.00	7.00

Table C.3 Parameters of the Speed-Governing and Turbine Systems

	Test Machine	Equivalent Machine
T_s	0.10	0.10
T_r	10.3	10.3
T_w	2.20	1.90
σ	0.05	0.05
δ	0.32	0.40
$(dG/dt)_{\min}$	–0.10	–0.10
$(dG/dt)_{\max}$	0.10	0.10
$G_{\min}$	0.00	0.00
$G_{\max}$	1.10	1.10

APPENDIX D TSSP BLOCK LIBRARY